

AD-A127 801

EXPERIMENTAL AND THEORETICAL INVESTIGATIONS OF FLOW
FIELDS AND HEAT TRANS. (U) MASSACHUSETTS INST OF TECH
CAMBRIDGE GAS TURBINE AND PLASMA D. E E COVERT ET AL.

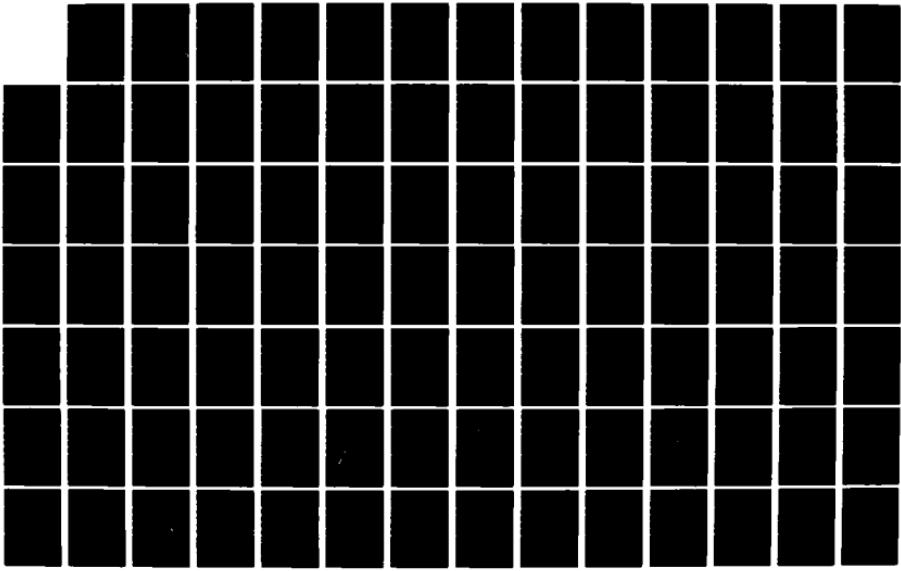
1/2

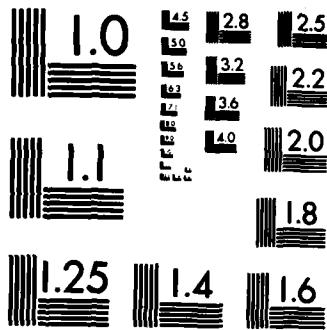
UNCLASSIFIED

31 DEC 81 N00014-81-K-0024

F/G 20/4

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

GAS TURBINE AND PLASMA DYNAMICS LABORATORY
DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS 02139

ADA 127801

ANNUAL REPORT NO. 1

on

81

ONR CONTRACT #N00014-K-0024

^

entitled

EXPERIMENTAL AND THEORETICAL INVESTIGATIONS OF FLOW FIELDS
AND HEAT TRANSFER IN MODERN GAS TURBINES

prepared for

DEPARTMENT OF THE NAVY
OFFICE OF NAVAL RESEARCH
ARLINGTON, VA 22217
ATTN: MR. KEITH ELLINGSWORTH
MECHANICS DIVISION

PERIOD OF
INVESTIGATION: JANUARY 1, 1981 - DECEMBER 31, 1981

PRINCIPAL INVESTIGATORS:

PROF. EUGENE E. COVERT
PROF. ALAN H. EPSTEIN
PROF. WILLIAM T. THOMPKINS, JR.

DTIC
SELECTED
S MAY 9 1983 D
A

DTIC FILE COPY

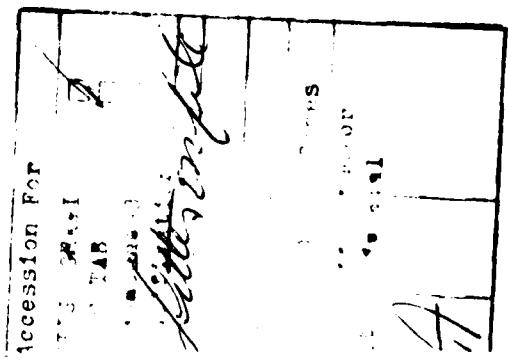
83 04 26 104

This document has been approved
for public release and sale; its
distribution is unlimited.

COMPUTATIONAL EFFORTS - INTRODUCTION

During the first year of the joint ONR-RR turbine project, computational efforts have concentrated on the areas of minicomputer-array processor development and two-dimensional computer code algorithm development. This annual report is intended to be a comprehensive review of the work performed in these areas as well as an analysis as to what the information gained during this year implies for the future direction of the project. The particular areas covered are: 1) a detailed review of the minicomputer-array processor performance and prospects, 2) a summary of the status and importance of boundary treatments for implicit numerical algorithms, 3) a summary of the status of grid generation efforts for turbine type geometries and 4) a review of work done to analyze the application of MacCormack's new implicit time marching scheme to turbomachinery geometries. In addition a computational development plan for the next three years is presented, and the present status of test examples for the two-dimensional analysis code is reviewed.

The minicomputer-array processor review largely appeared in an earlier quarterly report and is repeated here for completeness. The work on implicit boundary treatments was presented at a boundary condition symposium conducted at NASA Ames Research Center in June 1981 and an expanded version is expected to be published in the Journal of Computational Physics. The analysis of the new MacCormack scheme was presented at the AIAA 20th Aerospace Sciences Meeting in January 1982 as AIAA Paper 82-0063 and has been submitted to the AIAA Journal for publication. Technical papers reviewing the computer analysis and the results of the grid generation scheme are under preparation.



MINICOMPUTER-ARRAY PROCESSOR PERFORMANCE

Summary

One important portion of the original ONR-RR turbine project proposal was the development of a minicomputer-array processor system with sufficient memory and processing capacity to support the entire computational effort. Since the success of the computational project largely depended on the development of this computer system, it was imperative to establish early on the capacity of the purchased system. This report section reviews the development of this system and demonstrates that the original system computational speed goals have been met. The original computer concept used host computer main memory to provide storage for the three-dimensional problem data matrix, but the widespread introduction of the 64K bit memory chip has made it possible to specify a backup mass memory system which greatly expands the storage capacity while reducing its cost.

Historical Perspectives

It is easy to understand why existing general-purpose computers lack sufficient computing power for three-dimensional computational fluid dynamics calculations. General-purpose computers are based on a von Neumann type architecture, in which (a) a single processor is used, (b) a single, separate memory is used, and (c) program and data are stored in the same memory so that, at least in principle, programs can write other programs. These principles were formulated at a time when both arithmetic and logic units (ALU) and memory were expensive (hence only one of each is used, and data and programs can share a single address space), but data communication on internal bus lines was cheap (so that there was little if any penalty in separating the processor and memory). The large expense of high-performance machines made it imperative that all machines be general purpose, that is,

be able to perform well on a large variety of algorithms. However, these von Neumann architectures are all limited by two bottlenecks: the single-sequence bottleneck and the memory reference bottleneck.

The single-sequence bottleneck arises because only a single ALU or processor is incorporated. This restriction makes programming easier (indeed it has influenced the very nature of virtually all widely used computer languages), but it forces computations that could in principle be done simultaneously, to be done one after another. Heroic efforts have been made to eliminate this bottleneck, and techniques in common use in high performance computers include simultaneous fetch of next instruction from memory, and pipelined high-speed arithmetic operators. These approaches offer higher computation speed at the expense of greater hardware cost and complexity. A few machines have attempted to circumvent this bottleneck by incorporating two or more ALU's. In most cases such machines have given disappointing performance when more than four ALU's have been used, since having n ALU's does not reduce computation time by a factor of n . This is because techniques for unfolding or ("vectorizing") algorithms are only partially successful, and the ultimate computational speed is determined by those instructions that are not done in parallel. Thus even a relatively innocuous statement that, on a single-sequence machine, would not take much time, may end up on a vector machine being done on one ALU while all the others are idle. Perhaps the most successful vector machine is the CRAY-1, and in that case much of the success is due to the speed of the individual ALU's rather than the parallelism. If a vectorizing compiler is used, the CRAY-1 can routinely achieve 30 MFLOPS (million floating-point operations per second), which is considerably below

its potential of about 150 MFLOPS. In order to get maximum performance from the CRAY-1, a programmer must be aware of the detailed timing assumptions in the architecture, and then program so as to use many functional units concurrently. He must also use the concept of chaining vector operations, and deal with the fact that the preferred vector length is 64.

The memory reference bottleneck of von Neumann type computers arises because a single memory is used. Both instructions and data are fetched from or written to this memory. Typically the bandwidth of the memory communication path is relatively small, either because of slow memory access time or long distance between the memory and the ALU. In addition, the ALU may have to remain idle until the memory fetch is complete. A variety of successful techniques has reduced the impact of this bottleneck in high-performance computers. First of all, the principle of space locality is used to predict what memory locations will require access next. In practical problems a large percentage of memory accesses occur very close to the previous access. Thus if large segments of memory (pages) are brought into relatively fast memory, in a large majority of cases the next memory location required will already be in "fast" memory. The techniques of cache memory and paging are based on this principle and are very successful. In fact, practical computer systems have a memory hierarchy with at least four levels--magnetic tape (quite slow, massive size, used for day-to-day storage); disk (faster but smaller, used for most general purpose user-specified storage); cache (much faster, quite small, not usually under user control); and ALU registers (very fast, not many of them, located close by, used for storage of data from one instruction to the next).

Minicomputer-Peripheral Arithmetic Processor Concept

The design of a fluid dynamics simulator of sufficiently high power is quite difficult within the constraints imposed by these bottlenecks, and a common thread of the unconventional architecture concepts which have been considered is the attempt to avoid the von Neumann constraints by introducing parallelism on a modest or massive scale. In the minicomputer-peripheral arithmetic processor concept, a general purpose minicomputer is used to manage a memory hierarchy and data flow for a special purpose, high capacity ALU. A typical minicomputer-peripheral processor layout is illustrated in Figure 1. Here a general purpose minicomputer with substantial main memory and large moving head disk drives supports the peripheral arithmetic processor. Present computer hardware cost trends indicate that the host minicomputer should be a 32 bit supermini [Digital Equipment Corporation VAX 11/780 or Perkin Elmer Corporation 3240 for example] with one to two million bytes of main memory. Smaller and less expensive computers could also serve as host machines, but the most cost effective choice appears to be a 32 bit supermini.

The host minicomputer chosen for evaluation was a 32 bit minicomputer produced by Perkin Elmer, the PE 3242. Important architecture features of this machine are: 1) a high data rate internal memory bus, up to 10 million 32 bit words/second transfer rate; 2) four DMA (direct memory access) ports, 2.5 million words/sec transfer rate on each port; and 3) a cache memory system well suited to internal data shuffle operations. The effective memory access time is 500 ns, the cache cycle time is 200 ns, and the CPU cycle time is 260 ns. Floating point hardware multiply times are approximately 1 μ s for 32 bit single-precision results. A block diagram of the P.E. 3242 is shown in figure 2.

The peripheral arithmetic processor chosen for evaluation was the AP-120B array processor produced by Floating Point Systems, Inc. of Oregon. Floating Point Systems is the dominant manufacturer of such peripheral processors, and the AP-120B internal architecture is an excellent example of the potential computational advantages of abandoning the von Neumann style architecture. The AP-120B consists of a number of synchronous, parallel logical units, each operating under stored program control. A block diagram of the machine is shown in Figure 3.

The parallel structure of the AP-120B allows the overhead of array indexing, loop counting and data fetching from memory to be performed simultaneously with arithmetic operations on the data. Stored programs and data each reside in separate, independently addressable memories to reduce memory accessing conflicts. Independent floating point multiply and adder pipelines both allow operations to be initiated every machine clock cycle or 167 ns. Address indexing and counting functions are performed by an independent integer arithmetic unit. For certain computations, such as a Fast Fourier Transform, the computation rate is near that of a floating point multiply and add result every clock cycle or 12 MFLOPS (millions of floating point operations per second). The floating point data word is 38 bits long.

The AP-120B is connected to a PE3242 DMA port and this processor contains 2048 (2K) words of program storage memory and 32K words of 333 ns cycle time main data memory. The AP-120B cannot be considered a general purpose scalar or vector arithmetic processor. Achieving good performance requires careful, custom hand coding of critical code sections, and the limited function unit parallelism constrains the type of algorithm that can effectively use the machine. For example, a memory read instruction

may be initiated only every third machine cycle so that true vector operations like those available on the CDC STAR-205 are not possible. References to the floating point register file (data pad X and Y) are not sufficiently flexible to overcome this difficulty. To optimize the operation of the AP-120B, it is necessary for the programmer to "look ahead" and initiate memory reads prior to the actual time values from data memory are to be used in a calculation. The burden on the programmer to be cognizant of machine architecture is not greatly different from that of programmers on the CRAY or CDC STAR computer.

Numerical Solution Scheme

In order to properly focus the AP-120B architecture studies, the system was evaluated for use with a demanding algorithm, in terms of memory size and complexity, which might be suitable for solving the three-dimensional, unsteady Reynolds averaged Navier-Stokes equations. The numerical method is one due to Beam and Warming [1] of NASA Ames Research Center and may be generally described as an approximate factorization scheme for vector sets of convection-diffusion equations. This scheme may be represented as:

$$(I + L_x)(I + L_y)(I + L_z)\Delta U^n = RHS^n \quad (1)$$

In this representation, I is the identity matrix, L_x , L_y , L_z are linear, finite difference operators in the x-y-z coordinate directions, ΔU^n is the change in solution state vector from time step n to time step n+1, and RHS^n is, in effect, the steady-state solution to the Navier-Stokes equations at time level n. The computational solution proceeds as:

$$(I + L_x) \Delta U^* = RHS^n \quad \text{Step 1} \quad (2)$$

$$(I + L_y) \Delta U^{**} = \Delta U^* \quad \text{Step 2} \quad (3)$$

$$(I + L_z) \Delta U^n = \Delta U^{**} \quad \text{Step 3} \quad (4)$$

$$U^{n+1} = U^n + \Delta U^n \quad \text{Step 4} \quad (5)$$

where steps 1, 2, and 3 involve calculation of parameters, evaluation of the RHS, and solution of a large set of sparse, simultaneous, linear equations. The equation set to be solved is block tridiagonal and is solved by LU decomposition. The size of I and other submatrices appearing is $(m \times m)$ where m is the number of vector equations to be solved, $m = 5$ for ordinary problems. The number of simultaneous equations is m times the number of finite different nodes along a physical direction. Details of the finite difference algorithm are presented in reference [2].

The characteristic operation of this scheme is the solution of a large set of simultaneous equations, shown below in their banded matrix form.

$$\begin{array}{cccccc|c|c} A_0 & B_0 & C_0 & 0 & 0 & \dots & 0 & \Delta U_0 & D_0 \\ A_1 & B_1 & C_1 & 0 & 0 & \dots & 0 & \Delta U_1 & D_1 \\ 0 & A_2 & B_2 & C_2 & 0 & \dots & 0 & \Delta U_2 & D_2 \\ \vdots & \vdots & & & & & 0 & \vdots & \vdots \\ 0 & 0 & \dots & A_{J-1} & B_{J-1} & C_{J-1} & \Delta U_{J-2} & D_{J-2} \\ \vdots & \vdots & & & & & \Delta U_{J-1} & D_{J-1} \\ 0 & 0 & \dots & A_J & B_J & C_J & \Delta U_J & D_J \end{array} = \quad (6)$$

Here the block matrix elements, A_j , B_j , C_j , are themselves $m \times m$ whose elements depend only on the solution at time level n . D_j is a vector of length m that depends either on the solution at time level n or is known from the previous approximation step.

When using a von Neumann style computer having a single ALU and a limited amount of local register storage, the sequence of operations to find a solution is immaterial, and it may be found by:

1) Computing the RHS function of equation (2) at all node points and storing in main memory.

2) Solving the equation system (2), for each x coordinate line.

This operation requires that matrix coefficients A_j , B_j and C_j be calculated and a matrix equation like equation (6) be solved. The equation system is solved in two sweeps by LU decomposition. The forward sweep appears as:

$$E_j = G_j^{-1} (-B_j) \quad (7)$$

$$F_j = G_j^{-1} (D_j - C_j F_{j-1}) \quad (8)$$

$$G_j = C_j E_{j-1} + A_j \quad (9)$$

and the backward sweep would appear as:

$$\Delta U_j = E_j U_{j+1} + F_j \quad (10)$$

where G^{-1} indicates the inverse of G_j . Of course, for numerical calculations on a conventional computer, G_j^{-1} should never be found; rather, the elements of E_j are solved for directly. With certain types of computer hardware, it may be advantageous to compute G_j^{-1} .

3) Solving the matrix equations (3) for each y coordinate grid line as in step 2.

4) Solving the matrix equations (4) for each z coordinate grid line as in step 3 and updating the solution from time level n to time level $n + 1$.

A reference problem with a finite difference grid of $50 \times 50 \times 100$ will be considered. At each grid node about 20 floating point words must be

stored for a problem data storage requirement of 5.0 million words. For purposes of total timing estimates, it was assumed that 500 iterations or time steps would be required to compute a steady state solution.

It is felt that the solution time per iteration for the Beam and Warming algorithm is typical of many other algorithms and that the block tridiagonal matrix structure is typical of many implicit solution schemes. Thus solution times for this algorithm should be a good guide for hardware performance on other algorithms.

Two Dimension Test Computation - Host Computer Only

The Beam and Warming algorithm described in the previous section was first tested on the host minicomputer in a FORTRAN only version. For the two-dimensional test solution, the finite difference operator sequence appears as:

$$(I + L_x) \Delta U^* = RHS^n \quad \text{Step 1} \quad (11)$$

$$(I + L_y) \Delta U^n = \Delta U^* \quad \text{Step 2} \quad (12)$$

$$U^{n+1} = U^n + \Delta U^n \quad (13)$$

The performance of the present FORTRAN code is illustrated in figure 4 for a finite difference grid line of length 50 nodes. Such grid line requires 1.65 ms/node to compute the RHS function, 0.632 ms/node to compute the matrix coefficient and 2.17 ms/node to solve the matrix equation. A test problem of size 50 x 100 nodes would require 36 seconds per time step and 5 hours for 500 time steps.

Array Processor Test Calculation

The most difficult computational part of the test calculation is the calculation of the coefficients and the matrix equation solution. This portion of the calculation was moved to the array processor, and the result

is illustrated in figure 5. The RHS function still requires 1.65 ms/node to compute, since it was left in the host computer, but the matrix solution now requires only 0.34 ms/node rather than 2.80 ms. This result is a speed-up factor of approximately 8.2. An overhead factor of approximately 6 ms/line is incurred. With the RHS function calculated in the host computer, a time step now requires 20 seconds, and 500 time steps requires 2.7 hours. A total speed-up factor of 3.2 was achieved with just the coefficient and matrix equation solution done in the array processor. When the RHS calculation is moved into the AP-120B, it is estimated that a time step will require 5.46 seconds, and 500 time steps will require 45 minutes.

AP-120B Architecture Considerations

The first test calculations demonstrated both the advantages and the restrictions of the AP-120B architecture. These limitations are best explained by considering a vector operation which occurs often in CFD simulations:

$$\hat{U}_j^{n+1} = \hat{U}_j^n + \alpha_j (\hat{U}_{j+1}^n - \hat{U}_{j-1}^n); \quad j = 1 \text{ to } J \quad (14)$$

A vector of length J is updated using a central difference operator multiplied by a variable coefficient. Figure 6 shows how this operation could be carried out using multiple, chained pipelines and multiple register files. With such a configuration one final result, U_j^{n+1} , requiring 2 add operations and 1 multiply operation, can be obtained each machine clock cycle. The resulting computation rate is 18 MFLOPS (millions of floating point operations per second).

The AP-120B has only a single add unit and a single multiply unit and only a limited amount of register file storage, 64 words, so that chained pipeline operation is not possible. The best computation rate that can be

maintained on the example, equation (10), is 3 multiply operations and 6 add operations every 6 clock cycles. This computation rate is 9 MFLOPS.

The major architecture constraint of the AP-120B is the main data memory access time. Main data memory, or table RAM, cannot be simultaneously read from and written to, and only a single word can be read or written per clock cycle. The example algorithm requires 3 memory references per final result. For standard speed memory (333 ns cycle time) a memory reference can be initiated only every third clock cycle so that the example calculation is memory reference limited. Each final result requires 9 clock cycles for a computation rate of 2 MFLOPS.

In principle, the use of interleaved, fast memory (167 ns cycle time) would allow a 9 MFLOP computation rate to be maintained, but each memory reference address must be generated through S-Pad operations rather than through hardware. In practice, it appears that maximum computation rate that can be maintained is about 4 MFLOPS for general algorithms. Such a sustained computation rate is comparable to best CDC7600 speeds and is impressive for a machine whose hardware cost is around \$60,000.

An interesting result of the memory reference limitation is that the sequence of operation in the block LU decomposition, equations (7) through (10), can be critical. In this sequence, the number of temporary results, A_j , for example, scales as m^2 , where m is the number of conservation equations. If these coefficients, once calculated, were written to main data memory, the memory reference limitations would be much more important. The intermediate results, E_j and F_j , are needed later in the backward sweep, equation (10), and should be written to table RAM. In more general terms, effective use of the AP-120B requires the programmer to identify and exploit

local algorithm parallelism, concurrent calculation of coefficients (A_j , B_j , C_j) and storage of results (E_{j-1} and F_{j-1}) to minimize memory references.

Memory Hierarchy Considerations

The two-dimensional test calculation and architecture considerations showed that a large amount of AP-120B main data memory is not required to sustain calculations at the maximum plausible rates. Referring to figure 5, the overhead time needed to transfer data to and from the AP-120B is small and can be made negligible. The overhead results mainly from a poor Perkin-Elmer Floating Point Systems interface (hardware and software), and it appears that this overhead time can be reduced to less than 5% of computation time for problems of modest size.

If the AP-120B can work effectively on small amounts of data, problem data matrix storage can be maintained on a memory hierarchy rather than in the AP-120B main data memory. The hierarchy considered is to maintain the full data matrix (data matrix is of 4 to 40 million floating point words) on a mass storage device at all times; to maintain small sections of the data matrix, of order 150,000 floating point words, in the host computer memory for algorithm processing, and to transfer only a few thousand words to the AP-120B data memory for detailed computational tasks. For low cost systems, the mass storage device would be a moving head disk, while, for higher performance systems, the storage device would be a semi-conductor mass memory system.

For either mass storage device, segmentation of the problem matrix on the memory system is an important consideration; the segmentation considered is shown in figure 7. It is not known yet if this is the optimum segmentation,

but it is felt to be near optimum for small main memory size. The data base is arranged such that the smallest memory system addressable element is a $(5 \times 5 \times 5)$ node block containing $5 \times 5 \times 5 \times 20$ floating point words. The smallest working set to be in host computer memory is three axial data planes composed of 300 $(5 \times 5 \times 5)$ sub-blocks.

When an axial data plane set is in memory, the first two operators may be executed on one axial plane:

$$(I + L_x) \Delta U^* = RHS^n \quad (15)$$

$$(I + L_y) \Delta U^{**} = U \quad (16)$$

This axial plane must then be returned to the memory system and a new plane read in. When 100 axial planes have been processed, an axial grouping of $20 \cdot (5 \times 5 \times 5)$ blocks is read in and the last operator executed:

$$(I + L_z) \Delta U^n = U^{**} \quad (17)$$

$$U^{n+1} = U^n + \Delta U^n \quad (18)$$

One full iteration is completed when 100 sets of this last axial grouping have been assembled and processed.

This data structure implies a large quantity of data movement per solution time step since each axial plane must be moved from the memory system to the host computer memory to the AP-120B back to the host computer memory and back to the memory system. This process must also be repeated for each time step, which means that for the reference grid 32 MWords must be moved for each solution time step. The host computer selected for evaluation, the Perkin Elmer 3242, is probably unique in a commercially available minicomputer in that its internal data base structure is capable of handling this data quantity at high speed. Under worst case conditions,

the data bus structure can support an aggregate data rate of 10 MWords per second. Thus the entire data shuffle operation need require only 4 seconds real time if the AP-120B and the memory system can supply and accept data at an adequate rate.

The AP-120B can perform block data transfers at the rate of 1 word each clock cycle or 6 MWords per second. However, it is expected that only a small amount of data will be transferred to or from the AP-120B at any one time, and the I/O time will be dominated by the overhead time to initiate a transfer operation. This overhead time is estimated to be about 2 ms per transfer for a total overhead time of 25 seconds per time step on the $50 \times 50 \times 100$ reference grid.

As can be seen from these simple considerations, the overhead time in setting up the AP-120B data transfers is 5 times larger than the host data transfer time. However, the overhead time does imply a minimum vector length, which does depend on the numerical algorithm complexity, below which no gain is made by using the AP. For the test algorithm, equation (14) and a 2 ms overhead, a minimum vector length of about 300 is required if the host computer can perform floating point operations in an average time of 2 μ s. A 2 μ s average computation time is quite generous for the host computer considered.

This type of simple operation count, however, seriously overestimates the importance of AP-120B overhead since computational fluid dynamics problems are characterized by the fact that coefficient calculations, a_j , in equation (14), are complex algebraic functions of the current solution vector. For example, it appears that the minimum time to perform the block tridiagonal inversion is about 0.2 ms per node for the two-dimensional case.

For the overhead to be 10% of the calculation time for just the block inversion requires a vector length of 100. If the calculation time for the coefficients A_j , B_j , C_j is also considered, the vector length for 10% overhead time falls to 50, and it falls to 25 if the RHS calculation is considered. For three-dimensional calculations, the minimum vector length for 10% overhead is as low as 5 or 10. It is also to be remembered that the 10% overhead time is about 1% of the time required to do the calculation on the host computer.

The introduction of 64K RAM VLSI memory chips has made the concept of large, fast, inexpensive backup memory systems very attractive. Several vendors now supply memory systems made up from these chips complete with power supplies, error correction and custom interfacing designed as moving or fixed head disk replacements. A particularly interesting system is the memory system 3000 sold by Motorola Inc. which contains a maximum of 32M Bytes in a single chassis. A block diagram of this system is shown in figure 8.

The 64K RAM chips are arranged on 16 memory cards which are individually connected to an internal memory bus. The memory address controlled (ACC) has parallel access to all 16 memory cards, thus making available on the user bus 16 72-bit words each 500 ns cycle. In block transfer mode, this structure results in a 64M Bytes per second transfer rate. In random access mode, the data transfer rate is 16M Bytes per second. The maximum transfer of the host computer interface channel is 10M Bytes per second so that the Motorola memory system can supply data in the random access mode faster than the host computer can accept it.

An overhead or setup time will be required to initiate a transfer data between the memory system and the host computer, but the total overhead

time should be small compared to the AP-120B overhead time. Memory system read/write requests will be initiated for fewer times than I/O setups for the AP since the memory system will be used to store the data blocks illustrated in figure 6. These blocks are broken down to individual grid lines to transfer to the AP.

Hardware Cost Estimate

The hardware costs for the host computer with 2M Bytes main memory, a 300M Byte moving head disk and DMAI interfaces is \$200,000. The hardware cost for AP-120B with 32K words main data memory (333 ns cycle time) and P.E. 3242 interface is \$65,000. The Motorola memory system will cost approximately \$90,000 for 16M Bytes storage and perhaps \$20,000 for the custom interface required. The total hardware cost for a minimal system is \$375,000.

Price--Performance Estimate for Reference Grid

The two-dimensional test codes allow a rough estimate of the system performance on the reference grid problem. The AP-120B calculation time without consideration for overhead time is estimated to be 0.4 ms per mode, based on the two-dimensional simulations, which yields a calculation time of 41 hours. Assuming a 10% overhead time gives a total computation time of 45 hours. When the two-dimensional test codes were constructed, the memory reference limitations of the AP-120B were not properly understood, and it is estimated that the calculation time can be reduced by a factor of two. This saving results from not storing the coefficients, A_j , B_j , C_j in equations (6) through (10) in main data memory. The estimated computation time on the reference grid is then 23 hours. The average computation rate is approximately 3 MFLOPS with a total hardware cost of \$265,000.

A minimum cost system would replace the semiconductor memory system with the moving head disk which increases the total computation time by introducing the average disk seek time and slower data transfer rate into the calculation. This overhead time is roughly estimated to 24 hours and is not easily overlapped with AP-120B calculation time. A total computation of 47 hours is estimated when storing the solution data matrix on disk.

A maximum performance system would use the P.E. 3242 base structure to drive 4 AP-120B units. Since all calculations along grid lines can be made independent, the total computation time can be reduced by a factor of four. The total computation time would then be about 6 hours. The average computation rate would be about 12 MFLOPS with a total hardware cost of about \$570,000. The spectrum of price-performance is illustrated in figure 9.

Analysis and Discussion

One major factor leading to exceptional performance of the minicomputer-array processor combination is a good match between operating speeds of different system devices when the mass memory system uses the 64K bit MOS memory chips. The critical weakness of present supercomputers is in main data memory costs and communication strategies. In these machines main memory is made up of fast but expensive ECL (emitter coupled logic) chips which still require complex communication strategies to generate enough memory access bandwidth to keep pace with arithmetic processing. At the reduced pipeline speeds of the AP-120B, low cost MOS memory chips with simple communication strategies easily provide the needed memory access bandwidth. The balanced minicomputer system cost is reduced to the point that dedicated systems can provide larger memory sizes than available supercomputers.

REFERENCES

1. Warming, R.E. and Beam, R.M.: On the Construction and Application of Implicit Factored Schemes for Conservation Laws. Symposium in Computational Fluid Dynamics, New York, April 16-17, 1977. SIAM-AMS Proceeding, Vol. 11, 1977.
2. Bush, R.H.: Time Accurate Internal Flow Equations of the Thin Shear Layer Equations. MIT GT&PDL Report No. 156, Feb. 1981.

MINICOMPUTER - PERIPHERAL PROCESSOR BLOCK DIAGRAM

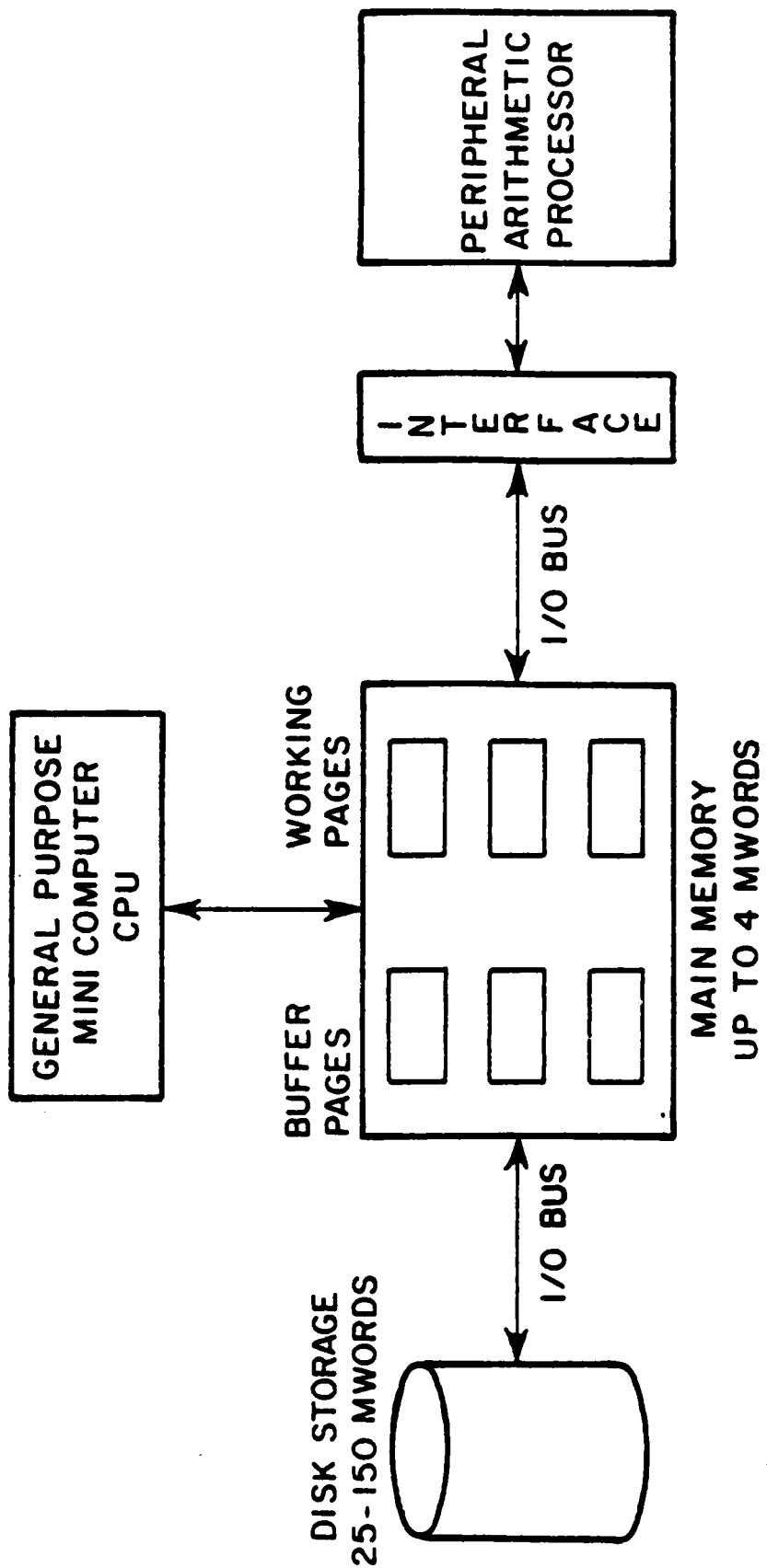


Figure 1

PERKIN- ELMER HOST MINI-COMPUTER
WITH PERIPHERAL PROCESSOR

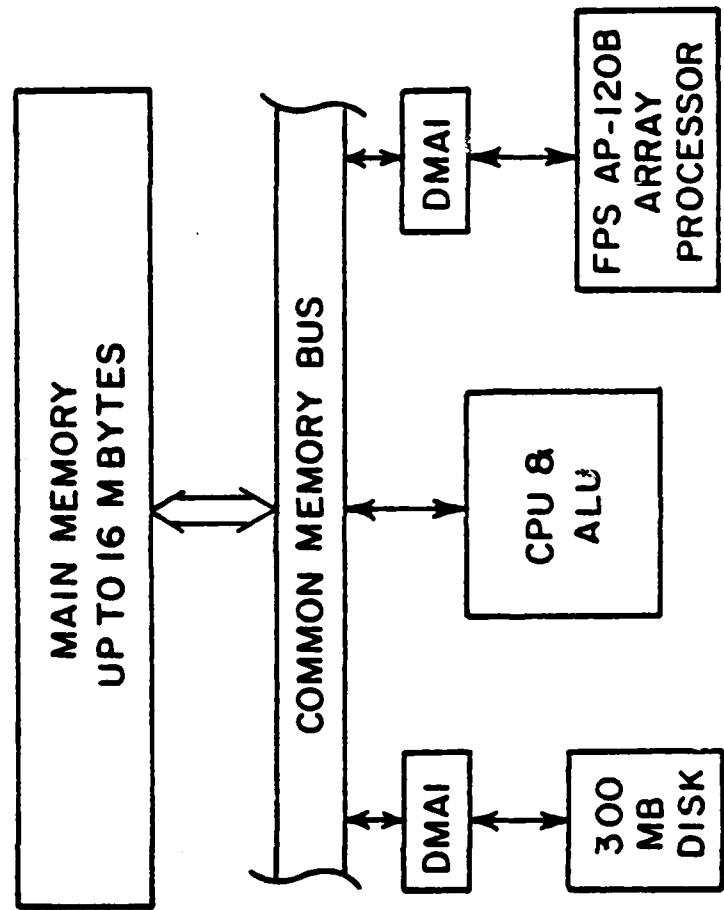
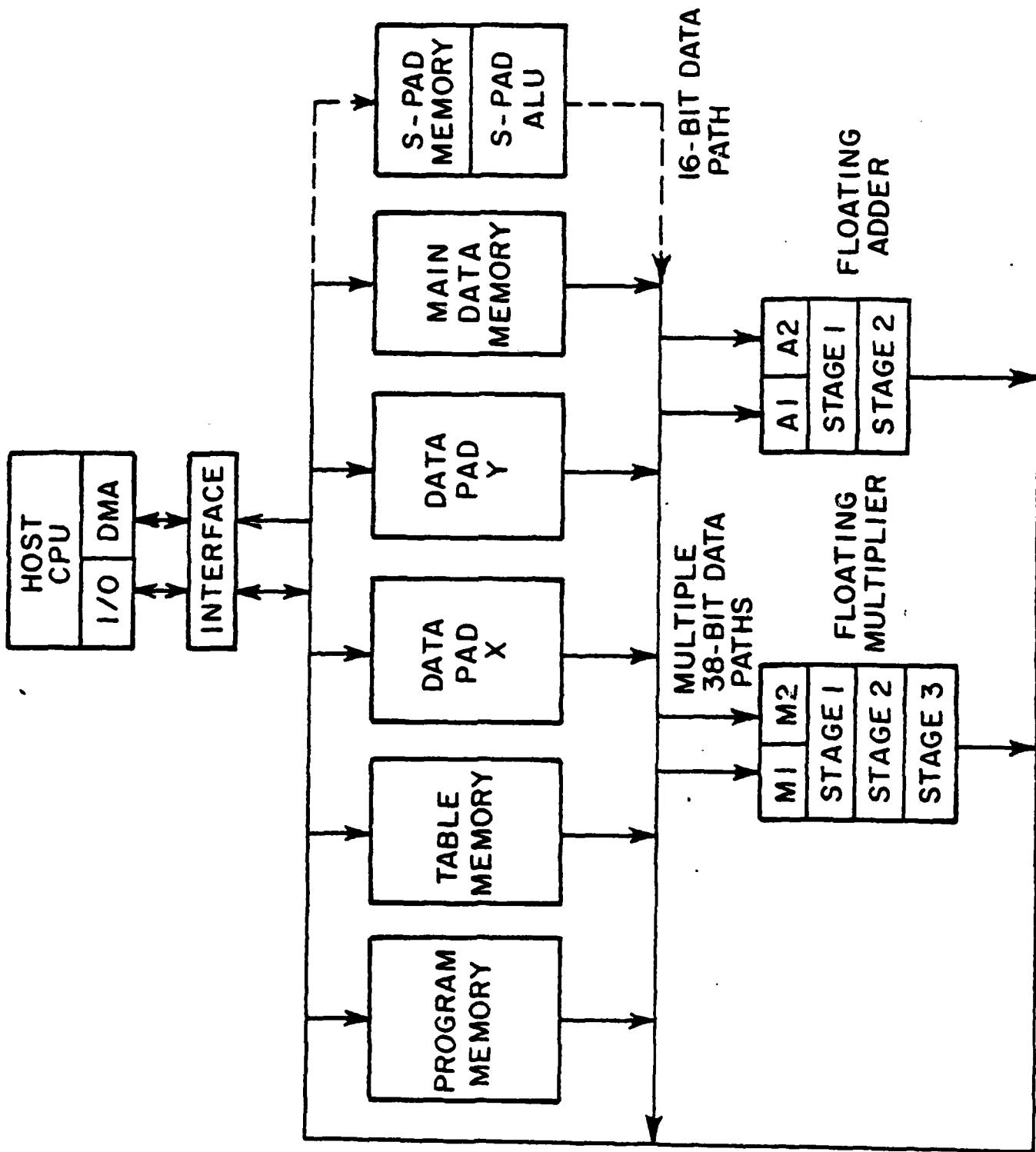


Figure 2

FLOATING POINT SYSTEMS AP-120B BLOCK DIAGRAM



P.E. 3242 TEST CASE EXAMPLE
SINGLE X SWEEP 50 NODES

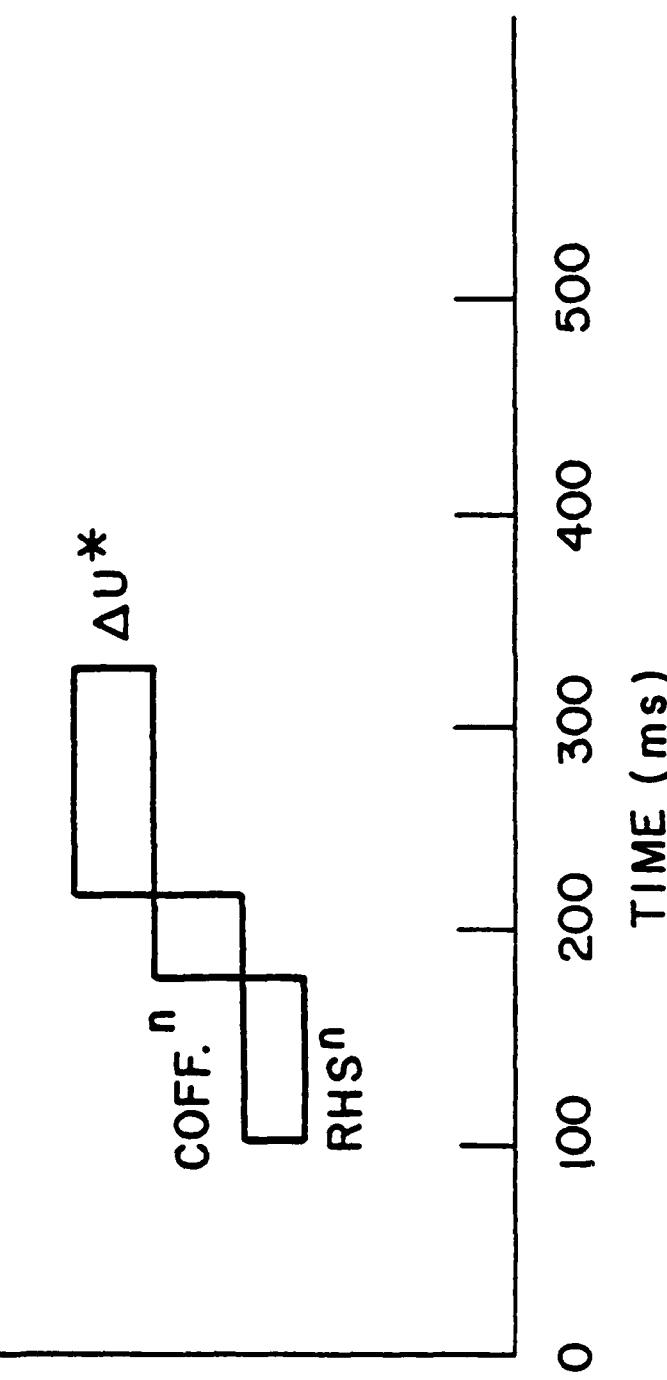


Figure 4

AP-120B TEST CASE
SINGLE X SWEEP 50 NODES

24

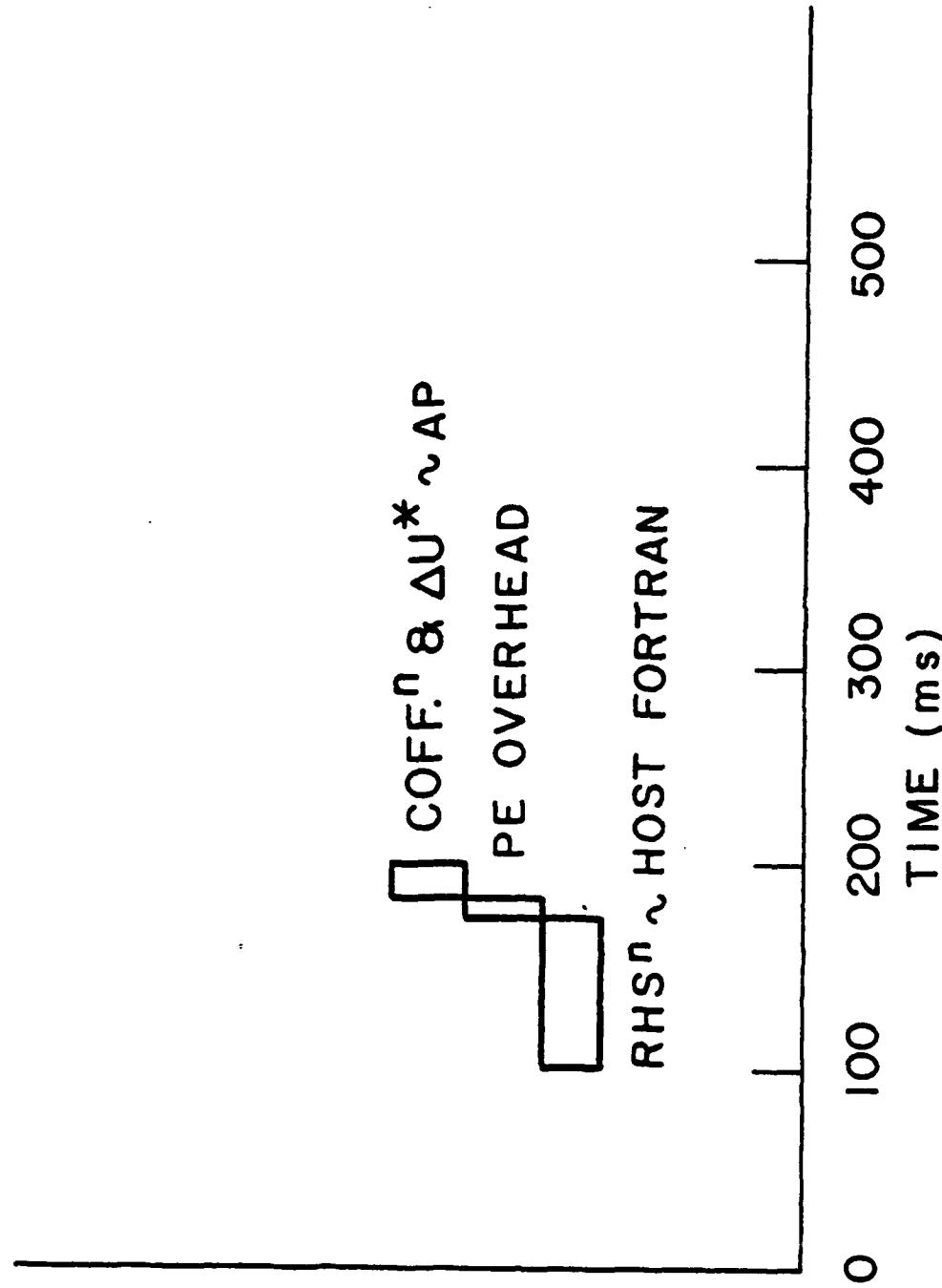


Figure 5

Figure 6

CHAINED PIPELINE PROCESSING FOR

$$U_j^{n+1} = U_j^n - \alpha_j (U_{j+1}^n - U_{j-1}^n)$$

Register File

U_1^n
U_2^n
U_3^n
\vdots
U_{11}
\vdots
U_{j-1}^n
U_j^n

Register File

U_1^n
U_2^n
U_3^n
\vdots
U_9
\vdots
U_{j-1}^n
U_j^n

Register File

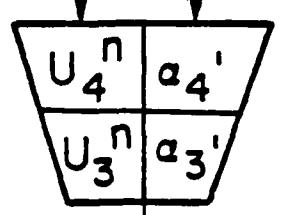
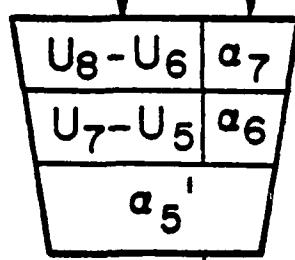
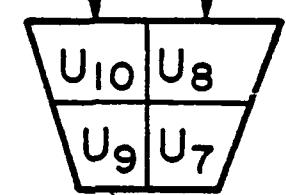
a_1
a_2
\vdots
a_8
\vdots
a_{j-1}
a_j

Register File

U_1^n
U_2^n
U_3^n
\vdots
U_5^n
\vdots
U_{j-1}^n
U_j^n

Register File

U_1^{n+1}
U_2^{n+1}
\vdots



PROBLEM DATA MATRIX STORAGE EXAMPLE

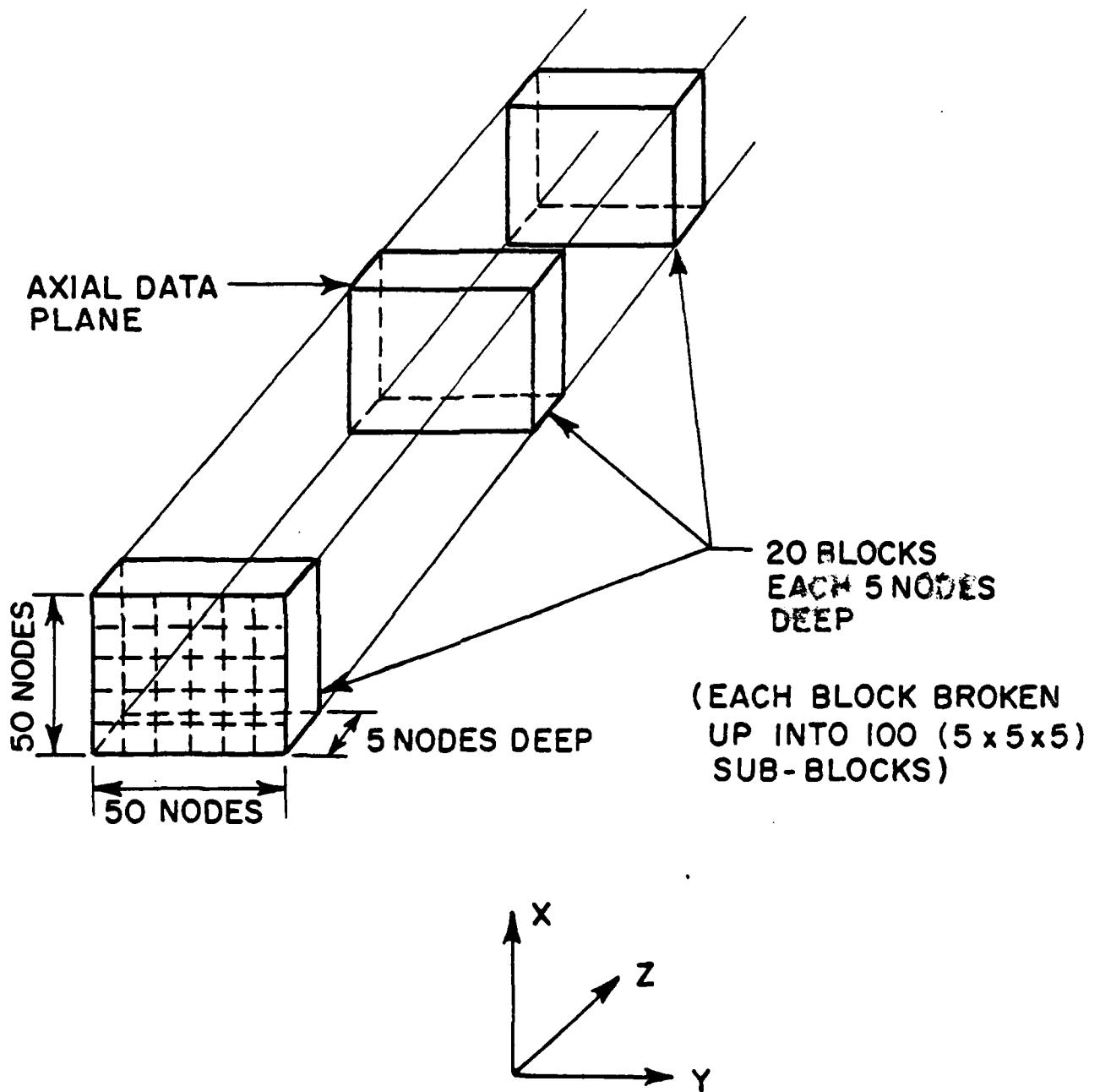
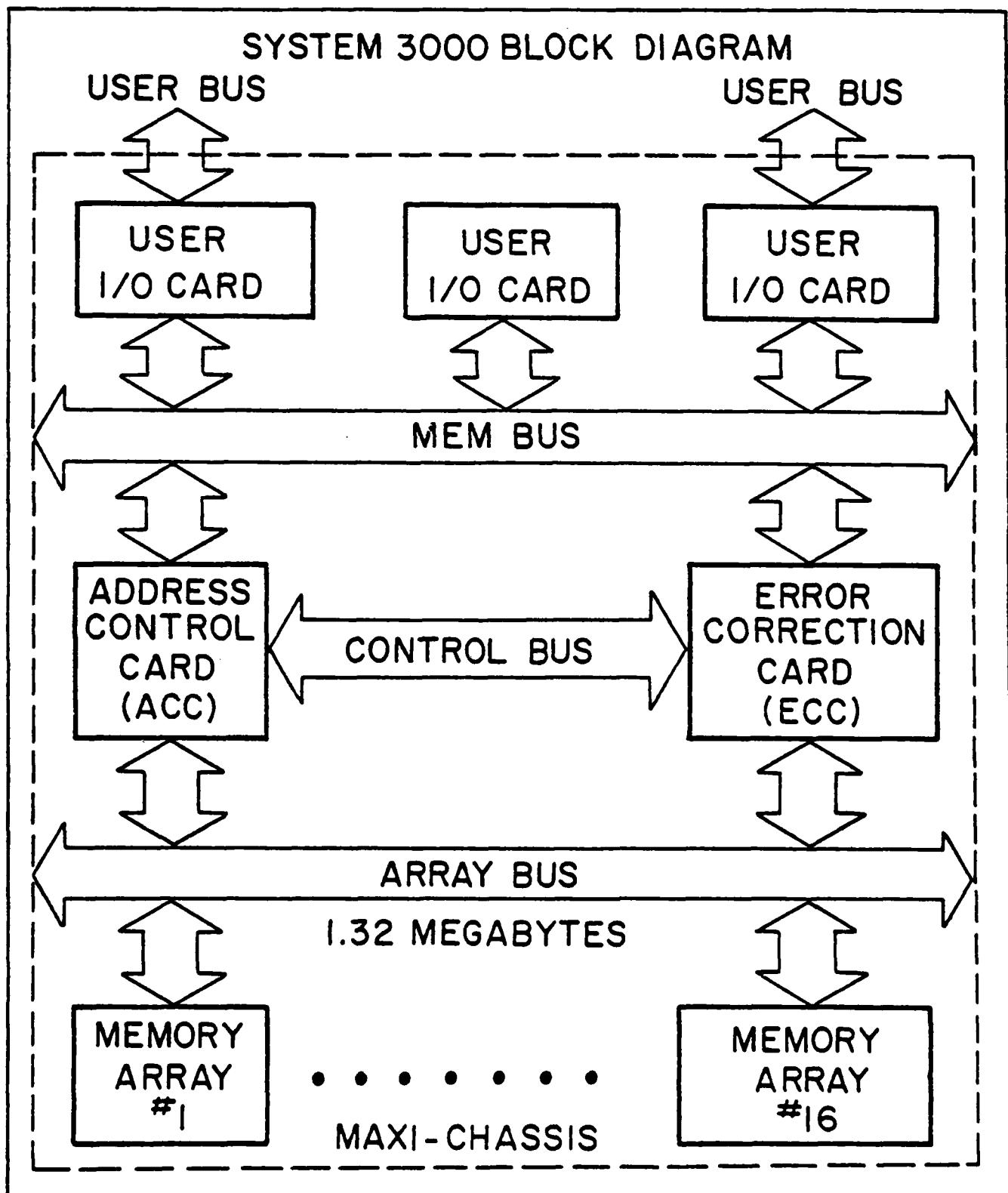
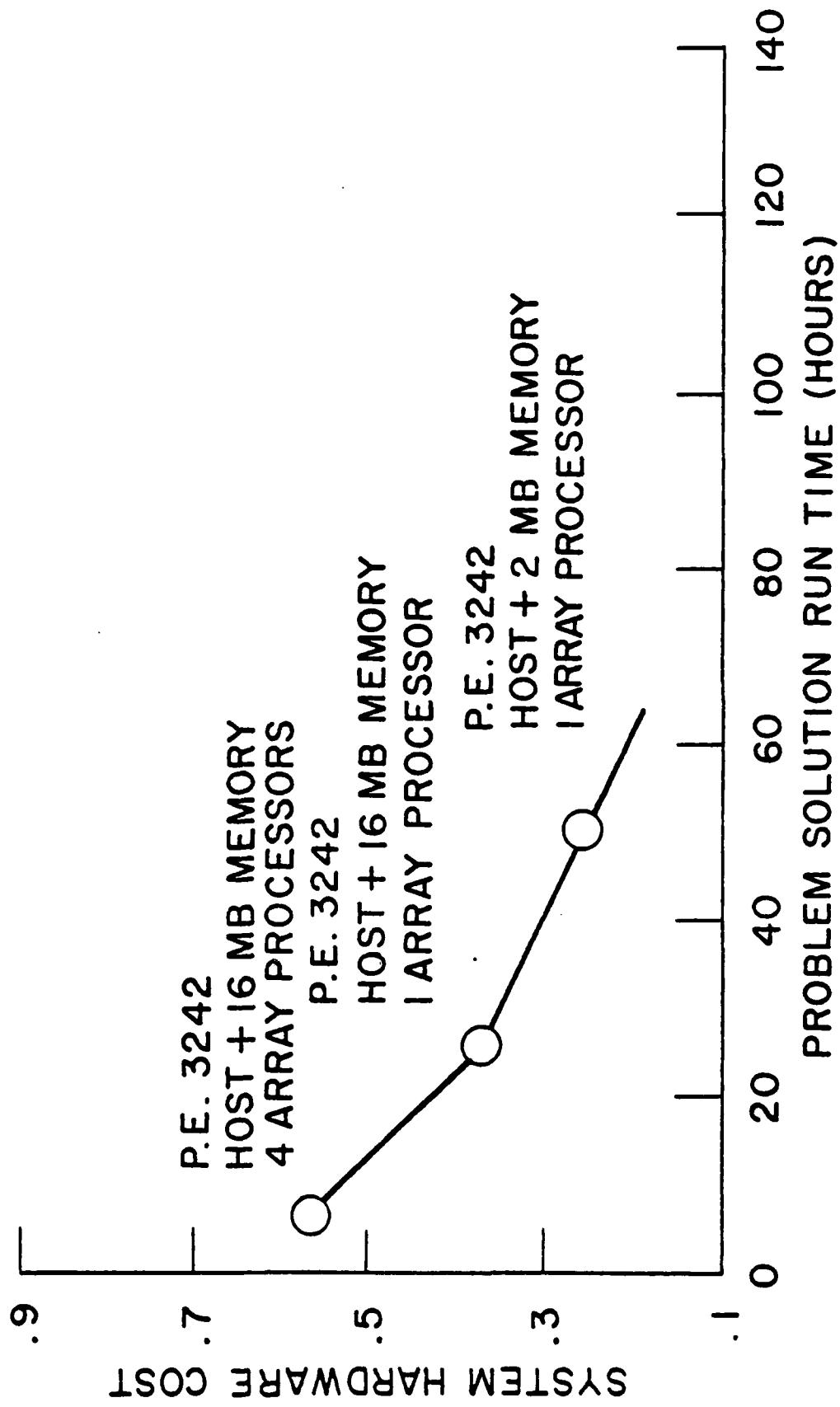


Figure 7

Figure 8
MOTOROLA MASS MEMORY BLOCK DIAGRAM



HARDWARE COSTS AND SOLUTION TIMES FOR 3-D VISCOUS
FLOW CODES USING ALGEBRAIC TURB. MODELING ON A
50 x 50 x 100 NODE POINT GRID FOR MINI-COMPUTER-
ARRAY PROCESSOR COMBINATIONS



BOUNDARY TREATMENTS FOR IMPLICIT SOLUTIONS TO EULER AND NAVIER-STOKES EQUATIONS

Summary

The importance of boundary treatments for implicit algorithms was greatly underappreciated until recent work by Yee, Beam and Warming [4] appeared. This work used a modal stability analysis originated by Kriess [5] to analyze the effect of several different boundary treatments on algorithm stability. This theory strictly applied only to linear equations with constant coefficient in one space dimension, and a computational study was conducted to test its relevance to realistic Euler or Navier-Stokes computations. It was found that for both explicit and implicit boundary treatments, it was possible to compute solutions with time steps 50 to 100 times explicit time limits while retaining the ability to choose rather arbitrary initial conditions. An even more important computation result which was observed is that while large time step sizes may be used, the largest convergence rates occur at relatively small time step sizes. For the two-dimensional test cases considered, the best time steps were of order 10 times the explicit limits. Since the present implicit codes require more than 10 times the operations per time step, the convergence rates of the implicit codes must be improved before they can be considered superior to the explicit algorithms.

BOUNDARY TREATMENTS FOR IMPLICIT SOLUTIONS
TO EULER AND NAVIER-STOKES EQUATIONS*

W. T. Thompkins, Jr.

R. H. Bush

Massachusetts Institute of Technology

INTRODUCTION

Implicit time marching schemes like those of Beam and Warming [1], Briley and McDonald [2], and MacCormack (1980) [3] generally have not been as robust as would be expected from a stability analysis for the pure initial value problem. Recently, Yee, Beam and Warming [4] illustrated that a more general stability analysis, which includes the effect of boundary conditions, may explain some of the seemingly anomalous behavior of these schemes. The major theoretical basis for this type of modal stability analysis was established in a series of papers by Kriess [5,6], Osher [7,8] and Gustafsson et al [9].

Yee as well as Gustafsson and Oliger [10] considered the effect of inflow/outflow boundary condition formulations on the stability of a class of numerical schemes to solve the Euler equations in one-space dimension. The characteristic feature of a subsonic inflow/outflow boundary is that a priori boundary values may be specified for only some problem variables while remaining boundary values must be determined as part of the solution process. Yee demonstrated a rather large disparity in stability bounds between the use of explicit or implicit extrapolation procedures and in general demonstrated that implicit extrapolation procedures had the least restrictive stability bounds. The intent of this work is to explore computationally the implication of this work for several two-dimensional Euler and Navier-Stokes simulations.

*This work was partially supported by NASA Lewis Research Center under NASA Grant NAG 3-9.

NUMERICAL PROCEDURES

The two-dimensional Navier-Stokes equations may be written in vector form as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} = \frac{\partial \mathbf{R}}{\partial x} + \frac{\partial \mathbf{S}}{\partial y}. \quad (1)$$

where

$$\begin{aligned}\mathbf{U} &= \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{pmatrix} & \mathbf{E} &= \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E_t + p) \end{pmatrix} & \mathbf{F} &= \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E_t + p) \end{pmatrix} \\ \mathbf{R} &= \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ R_4 \end{pmatrix} & \mathbf{S} &= \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ S_4 \end{pmatrix}\end{aligned}$$

and

$$E_t = \rho \left(e + \frac{1}{2} (u^2 + v^2) \right)$$

$$\tau_{xx} = (\lambda + 2\mu) \frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y}$$

$$\tau_{xy} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)$$

$$\tau_{yy} = (\lambda + 2\mu) \frac{\partial v}{\partial y} + \lambda \frac{\partial u}{\partial x}$$

$$R_4 = u \tau_{xx} + v \tau_{xy} + \mu p_r^{-1} (\gamma - 1)^{-1} \frac{\partial a^2}{\partial x}$$

$$S_4 = u \tau_{xy} + v \tau_{yy} + \mu p_r^{-1} (\gamma - 1)^{-1} \frac{\partial a^2}{\partial y}$$

The strong conservation law form may be retained under a general coordinate mapping as illustrated in Viviand [11]. All computations to be described were conducted in a mapped computational domain but numerical and boundary condition procedures will be described in the simple two-dimensional geometry shown in figure 1 for simplicity.

A 1979 paper by Beam and Warming [12] outlined a solution scheme for systems of equations of the form (1) which includes most numerical schemes for which the modal boundary condition analysis has been conducted. This scheme uses the well developed methods for ordinary differential equations as a guide to developing numerical methods for partial differential equations. The scheme presented combines Linear Multistep Methods, local linearization, approximate factorization and One Leg methods. The scheme, a generalization of the scheme presented in reference [1], solves for a variable $\rho(E)u$ which is equivalent to ΔU^n in the class of schemes represented by the earlier paper. The earlier scheme is somewhat easier to understand as ΔU^n is just the change in the solution from time level n to level $n+1$, while $\rho(E)u$ is a more general time differencing formula.

The solution schemes chosen are implemented as:

$$(I + L_y^n) \Delta U^* = RHS^n . \quad (2)$$

$$(I + L_x^n) \Delta U^n = \Delta U^* \quad (3)$$

$$U^{n+1} = U^n + \Delta U^n \quad (4)$$

where

RHS^n is very nearly the finite difference approximation to the steady state equations, and

L_x and L_y are linearized finite difference operators representing a particular time and spatial differencing scheme.

Full details of these operators are contained in Beam and Warming [1]. If the spatial differencing is taken to be centered, the computational form of either equation (2) or (3) appears at each interior point as:

$$A_i^n \Delta U_{i-1}^n + B_i^n \Delta U_i^n + C_i^n \Delta U_{i+1}^n = D_i^n \quad (5)$$

where A_i , B_i , and C_i are 4×4 matrices, known at time level n , D_i is the right-hand side vector at node point i known at time level n , and ΔU_i^n is the unknown vector at node point i . The boundary points will be assumed to involve only the nearest two points in the x direction.

$$A_0^n \Delta U_0 + B_0^n \Delta U_1 + C_0^n \Delta U_2 = D_0^n \quad (6)$$

The restriction to extrapolation along grid lines, actually transformed grid lines, is necessary to maintain the block tridiagonal form and avoids possible instabilities due to skewed extrapolation, see Abarbanel and Murman [13]. Extrapolation procedures using more than the two nearest neighbors can also be included in the process to be described.

The full matrix equation appears as:

$$\begin{array}{ccccccccc|c|c|c} A_0 & B_0 & C_0 & 0 & 0 & \dots & 0 & \Delta U_0 & & D_0 \\ A_1 & B_1 & C_1 & 0 & 0 & \dots & 0 & \Delta U_1 & & D_1 \\ 0 & A_2 & B_2 & C_2 & 0 & \dots & 0 & \Delta U_2 & = & D_2 \\ 0 & 0 & & & & & 0 & \Delta U_{n-2} & & D_{n-2} \\ \vdots & \vdots & & A_{n-1} & B_{n-1} & C_{n-1} & & \Delta U_{n-1} & & D_{n-1} \\ 0 & 0 & \dots & A_n & B_n & C_n & & \Delta U_n & & D_n \end{array} \quad (7)$$

and will reduce to tridiagonal form if the first and n^{th} equations are substituted into the second and the $n-1^{th}$ equations.

$$\left| \begin{array}{ccccc} B_1' & C_1' & 0 & \cdots & 0 \\ A_2 & B_2 & C_2 & \cdots & 0 \\ 0 & & & \vdots & \\ & & & 0 & \\ \vdots & & A_{n-2} & B_{n-2} & C_{n-2} \\ 0 & \cdots & 0 & A_{n-1}' & B_{n-1}' \end{array} \right| \left| \begin{array}{c} \Delta U_1 \\ \Delta U_2 \\ \vdots \\ \Delta U_{n-2} \\ \Delta U_{n-1} \end{array} \right| = \left| \begin{array}{c} D_1' \\ D_2 \\ \vdots \\ D_{n-2} \\ D_{n-1}' \end{array} \right| \quad (8)$$

where for example

$$B_1' = B_1 - A_1 A_0^{-1} B_0.$$

This cumbersome development allows us to show clearly how a large variety of explicit or implicit boundary forms can be included without difficulty.

BOUNDARY TREATMENTS

Inflow/Outflow Boundary

The finite difference algorithms studied usually require more boundary values than are required for the partial differential equations which they simulate. These extra numerical boundary conditions cannot be set arbitrarily and are usually determined through an extrapolation procedure. These extrapolation procedures may either be explicit, that is boundary values needed at a new time are determined uniquely from the old time level solution, or implicit, that is boundary values are determined as part of the new time level solution. The analytical boundary conditions or the extrapolation quantities are usually not conservation variables but primitive variables and a local linearization is usually required as part of defining the extrapolation procedure.

Consider, for example, an implicit, subsonic outflow boundary at which the local static pressure is specified as a boundary condition, and all other variables are to be determined by extrapolation. Figure 1 shows a typical computational grid and defines the subscripts used.

$$p_{i,j}^{n+1} = p_{i,j}^n ; \text{ given} \quad (9)$$

$$\begin{bmatrix} \rho \\ \rho u \\ \rho v \end{bmatrix}_{i,j}^{n+1} = 2 \begin{bmatrix} \rho \\ \rho u \\ \rho v \end{bmatrix}_{i,j-1}^{n+1} - \begin{bmatrix} \rho \\ \rho u \\ \rho v \end{bmatrix}_{i,j-2}^{n+1} \quad \begin{array}{l} \text{implicit} \\ \text{space} \\ \text{extrapolation} \end{array} \quad (10)$$

In order to complete the boundary formulation, all equations must be expressed in delta form and in terms of conservation variables. For the total internal energy this may be done through its definition:

$$E_t = \frac{P}{(\gamma-1)} + \frac{1}{2} \left(\frac{(\rho u)^2}{\rho} + \frac{(\rho v)^2}{\rho} \right) \quad (11)$$

Since the relations between conservation variables are nonlinear, some linearization step will be necessary before the boundary condition formulation may be used. We choose to introduce our linearization step here as:

$$\begin{aligned} \Delta E_t &= (E_t^{n+1} - E_t^n) = \frac{1}{(\gamma-1)} \Delta P - \frac{1}{2} (u^n)^2 \Delta \rho + u^n \Delta (\rho u) \\ &\quad + v^n \Delta (\rho v) + (\Delta u \Delta v, \Delta u^2, \Delta v^2, \Delta \rho \Delta u, \Delta \rho \Delta v) \end{aligned} \quad (12)$$

If terms of order $\Delta u \Delta v$ are neglected, the error is equivalent to the linearization error of the interior point scheme. We may express the transformation from boundary variables to conservation variables as:

$$\begin{bmatrix} \Delta \rho \\ \Delta \rho u \\ \Delta \rho v \\ \Delta E_t \end{bmatrix}_{i,j} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{-(u^n)^2}{2} & u^n & v^n & \frac{1}{(\gamma-1)} \end{bmatrix} \begin{bmatrix} \Delta P \\ \Delta \rho u \\ \Delta \rho v \\ \Delta P \end{bmatrix} = N_{i,j} \Delta W_{i,j} \quad (13)$$

We will in general denote transformation from conservative to primitive variables as

$$\Delta W_{i,j} = T_{i,j} \Delta U_{i,j} \quad (14)$$

The extrapolation conditions for $\Delta W_{i,j}$ are:

$$\begin{bmatrix} \Delta \rho \\ \Delta \rho u \\ \Delta \rho v \\ \Delta P \end{bmatrix}_{i,J} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \rho \\ \Delta \rho u \\ \Delta \rho v \\ \Delta P \end{bmatrix}_{i,J-1} + \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \rho \\ \Delta \rho u \\ \Delta \rho v \\ \Delta P \end{bmatrix}_{i,J-2} \quad (15)$$

or

$$W_{i,J} = P_{J-1} W_{i,J-1} + P_{J-2} W_{i,J-2} \quad (16)$$

The final equations relating the boundary conservation variables and the interior conservation variables are:

$$\Delta U_{i,J} = N_{i,J}^n \left(P_{J-1} T_{i,J-1}^n \Delta U_{i,J-1} + P_{J-2} T_{i,J-2}^n \Delta U_{i,J-2} \right) \quad (17)$$

or

$$\Delta U_{i,J} = G_{i,J-1}^n \Delta U_{i,J-1} + H_{i,J-2}^n \Delta U_{i,J-2} \quad (18)$$

With the definition of P_{J-1} and P_{J-2} given in equation (15), $T_{i,J-1}$ and $T_{i,J-2}$ are identity matrices.

An explicit outflow boundary treatment was constructed using:

$$P^{n+1} = P^n ; \text{ given}$$

$$\begin{bmatrix} \rho \\ \rho u \\ \rho v \end{bmatrix}_{i,J}^{n+1} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \end{bmatrix}_{i,J-1}^n \quad (19)$$

and setting $G_{i,J-1} = H_{i,J-2} = 0$.

In forming equation (10), we choose to extrapolate the local momentum flux rather than a specific primitive or characteristic variable; choice of other extrapolation variables would alter only the transformation matrix, $T_{i,j}$. Extrapolation of the momentum flux is somewhat arbitrary, but its choice did not affect the accuracy of the computational results to be presented.

Solid Wall Boundary Procedures

The boundary treatment procedures illustrated for inflow/outflow boundary are easily extended to cover solid walls in either inviscid or viscous flow situations. Here,

$$\begin{bmatrix} \Delta U_{0,j} \\ \Delta P \\ \Delta \rho u \\ \Delta \rho v \\ \Delta E_t \end{bmatrix} = \begin{bmatrix} \Delta \rho & \gamma/T & \rho/T & 0 & 0 \\ \Delta \rho u & \gamma u/T & \rho u/T & 0 & 0 \\ \Delta \rho v & \gamma v/T & \rho v/T & 0 & S q \\ \Delta E_t & \frac{1}{\gamma-1} + \frac{1}{2} \frac{\gamma v^2}{T} - \frac{1}{2} \frac{\rho q}{T} & \rho q & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta P \\ \Delta T \\ \Delta q \\ \Delta u \end{bmatrix} \quad (20)$$

or

$$\Delta U_{0,j} = N_{0,j}^n \Delta W_{0,j} \quad (21)$$

where q is the velocity parallel to the wall and S is the wall slope. For the inviscid flow examples $\partial P/\partial y$, $\partial T/\partial y$ and $\partial U/\partial y$ are set equal to zero, while, for the viscous flow examples v , u , and $\partial T/\partial y$, are set equal to zero and $\partial P/\partial y$ is equal to $4/3 \mu (\partial^2 P / \partial y^2)(v)$. All derivatives are evaluated by one-sided finite difference formulas.

As indicated by Buggeln, Briley and McDonald [14], an ADI type procedure requires boundary conditions for the intermediate step. Usually the intermediate step was in the "y" direction and the boundary conditions were applied as if the intermediate results were physical quantities, that is, the boundary conditions of equation (20) were applied to the quantities ΔU^* of equation (2).

Explicit wall boundary treatments are generated by applying the primitive variable form of equation (20) and forcing the correction matrices to be zero.

NUMERICAL RESULTS

Three geometries were selected for detailed study: an inviscid super-sonic diffuser with weak oblique shock, supersonic in - supersonic out; an inviscid supersonic diffuser with a strong normal shock, supersonic in - subsonic out, and a viscous supersonic diffuser with weak oblique shock illustrating a shock-boundary layer interaction. Sketches of the geometries and ideal solutions are shown in figures 2, 3 and 4. Solutions for each geometry were run to steady state for a range of time step sizes. For convenience time step sizes are reported in terms of x and y CFL numbers:

$$(CFL_x) = \text{maximum} \left[\frac{\Delta t(u + c)}{\Delta x_{i,j}} \right] \quad (22)$$

$$(CFL_y) = \text{maximum} \left[\frac{\Delta t(v + c)}{\Delta y_{i,j}} \right] \quad (23)$$

The time step size was uniform over each calculation which results in non-uniform CFL_x and CFL_y numbers. The maximum value of each is reported. Sample convergence history plots are shown in figure 5 which shows the log of the value of the point maximum steady state residual

$$SSR = \frac{\partial E}{\partial x} - \frac{\partial R}{\partial x} + \frac{\partial F}{\partial y} - \frac{\partial S}{\partial y} \quad (24)$$

plotted against the iteration number. A solution was not termed stable unless the residual converged to the machine accuracy, about 1×10^{-6} . All calculations used a 32 bit floating point word size.

Each geometry calculation was run with fully explicit extrapolations, $\Delta U = 0$, and with fully implicit extrapolations, and the results summarized in Table 1. The most interesting of these results are shown in figure 5. At a time step size corresponding to a CFL_x number of 15 convergence was rapid and very nearly monotonic in time. At smaller time step sizes, the convergence was slower but nearly monotonic. At a CFL_x of 45 convergence rates initially appeared to be faster than for a CFL_x of 15, but the final residual values oscillated significantly about its minimum value. At a CFL_x of 90, the convergence rate was substantially slower than at a CFL_x of 15, and at larger CFL_x values the solution diverged.

The results for the strong shock diffuser can reasonably be compared to those of Yee, Beam and Warming [4]. They reported a CFL number stability limit between 10 and 20, while we found stability limits between 90 and 150. Thus the analysis in one-space dimension does appear to provide a sufficient condition for stability, but it may not provide a close approximation to the stability limit. However, it is essential to emphasize that the largest convergence rates were observed at time steps corresponding to CFL numbers of order 10 and that only a marginal computational time advantage for the implicit boundary formulations was observed.

The results for the shock-boundary layer calculation are very interesting, but they demonstrate a substantial computational advantage for the implicit solid wall conditions, not for the inflow/outflow extrapolation. Here the stability boundary and the best convergence rates were observed at time step sizes corresponding to CFL_x numbers of 5 to 10. When using the implicit wall conditions, the algorithm stability appeared to be independent of grid spacing in the normal direction as might be hoped. When using the explicit wall condition, the algorithm stability was limited to a CFL_y number of about 500.

CONCLUSIONS AND DISCUSSION

While it is difficult to generalize from only a few test examples, it is apparent that a better appreciation of the role boundary treatments play in implicit algorithms has allowed the development of far more robust Beam and Warming type solvers. For both explicit and implicit boundary treatments, we were able to accurately compute solutions with time steps 50 to 100 times explicit time limits while retaining the ability to choose rather arbitrary initial conditions. In many cases, our limiting time steps for the two-dimensional test problems were in fact larger than the limit which a one-dimensional analysis would suggest.

The most important computational result we observed was that while an improved appreciation of boundary treatments did allow very large time step sizes to be used, the largest convergence rates to steady state were observed at relatively small time step sizes. For the two-dimensional test problems, the best CFL_x numbers were of order 10, not of order 100. One-dimensional test examples showed no such convergence rate behavior. Presently unpublished analysis by Abarbanel, Dwoyer and Gottlieb [15] has linked this behavior to the approximate factorization form of equations 2 and 3. This effect now seems to be setting the time step sizes for our viscous flow computations and new work should focus on methods for overcoming this limitation.

Table 1

Inviscid Weak Oblique Shock Diffuser				Inviscid Strong Shock Diffuser				Inviscid Shock Boundary Layer			
CFL _x	CFL _y	Iterations to Convergence	CFL _x	CFL _y	Iterations to Convergence	CFL _x	CFL _y	Iterations to Convergence			
Explicit Boundary Treatments	1.5	0.6	225	1.5	0.6	3000	1.5	125	1300		
	6.0	2.4	130	6.0	2.4	1200	6.0	500	Unstable		
	15.0	6.0	225	15.0	6.0	800	15.0	1250	Unstable		
	45.0	18.0	500	45.0	18.0	800	45.0	3750	Unstable		
	90.0	36.0	1000	90.0	36.0	1000	90.0	7500	Unstable		
	150.0	60.0	3000	150.0	60.0	3000	150.0	12500	Unstable		
	1.5	0.6	225	1.5	0.6	3000	1.5	125	1300		
	6.0	2.4	130	6.0	2.4	950	6.0	500	400		
	15.0	6.0	225	15.0	6.0	400	15.0	1250	Unstable		
	45.0	18.0	500	45.0	18.0	1000	45.0	3750	Unstable		
Implicit Boundary Treatments	90.0	36.0	1000	90.0	36.0	1500	90.0	7500	Unstable		
	150.0	60.0	Unstable	150.0	60.0	Unstable	150.0	12500	Unstable		
	15.0	1250	250				15.0	600	Unstable		

NOTE: Backward Euler time differencing used for all reported calculations

REFERENCES

1. Beam, R.M. and R.F. Warming: An Implicit Factored Scheme for the Compressible Navier-Stokes Equations. AIAA Journal, Vol 16, No. 4, April 1978, pp 393.
2. Briley, W.R. and H. McDonald: Three Dimensional Supersonic Flow of a Viscous or Inviscid Gas. J.Comp.Phys., Vol 19, 1975, pp 150.
3. MacCormack, R.W.: A Numerical Method for Solving the Equations of Compressible Viscous Flow. AIAA Paper 81-0110, 1980.
4. Yee, H.C., R.M. Beam and R.M. Warming: Stable Boundary Approximations for a Class of Implicit Schemes for the One-Dimensional Inviscid Equations of Gas Dynamics. Proc. of AIAA Computational Fluid Dynamics Conference, Paper No. 81-1009, June 1981.
5. Kreiss, H.O.: Stability Theory for Difference Approximations of Mixed Initial Boundary Value Problems. I. Mathematics of Computations, Vol. 22, 1968, pp 703-714.
6. Kreiss, H.O.: Difference Approximations for Initial Boundary Value Problems. Proc. of the Royal Society of London, Series A, Vol 323, 1971, pp 255-261.
7. Osher, S.: Systems of Difference Equations with General Homogeneous Boundary Conditions. Transactions of the American Mathematical Society, Vol 137, 1969, pp 177-201.
8. Osher, S.: Stability of Difference Approximations of Dissipative Type for Mixed Initial Value Problems. I. Mathematics of Computations, Vol 23, 1969, pp 335-340.
9. Gustafsson, B., H.O. Kreiss and A. Sundstrom: Stability Theory of Difference Approximations for Mixed Initial Boundary Value Problems. II. Mathematics of Computations, Vol 26, 1972, pp 649-686.
10. Gustafsson, B. and J. Oliger: Stable Boundary Approximations for a Class of Time Discretizations of $U_t = AD_0 U$. Report No. 87, Dept. of Computer Science, Upsala U., Sweden, Sept. 1981.
11. Viviand, H.: Conservation Forms of Gas Dynamic Equations. La Recherche Aerospatiale, No. 1, Jan. 1974, pp 65-68.
12. Beam, R.M. and R.F. Warming: An Implicit Factor Scheme for the Compressible Navier-Stokes Equations. II: The Numerical ODE Connection. AIAA paper 79-1446, Williamsburg, VA, 1979.
13. Abarbanel, S.S. and E.M. Murman: Stability of Two-Dimensional Hyperbolic Initial Boundary Value Problems for Explicit and Implicit Schemes. Symposium on Numerical Boundary Conditions, NASA Ames Rsch Ctr, Oct. 1981.
14. Buggeln, R.C., W.R. Briely and H. McDonald: Solution of the Navier-Stokes Equations for Three-Dimensional Turbulent Flow with Viscous Sublayer Resolution. Proceedings of AIAA Computational Fluid Dynamics Conference, paper No. 81-1023, June 1981.
15. Abarbanel, S.S., D.L. Dwoyer and D. Gottlieb: Private communication, July 1981.

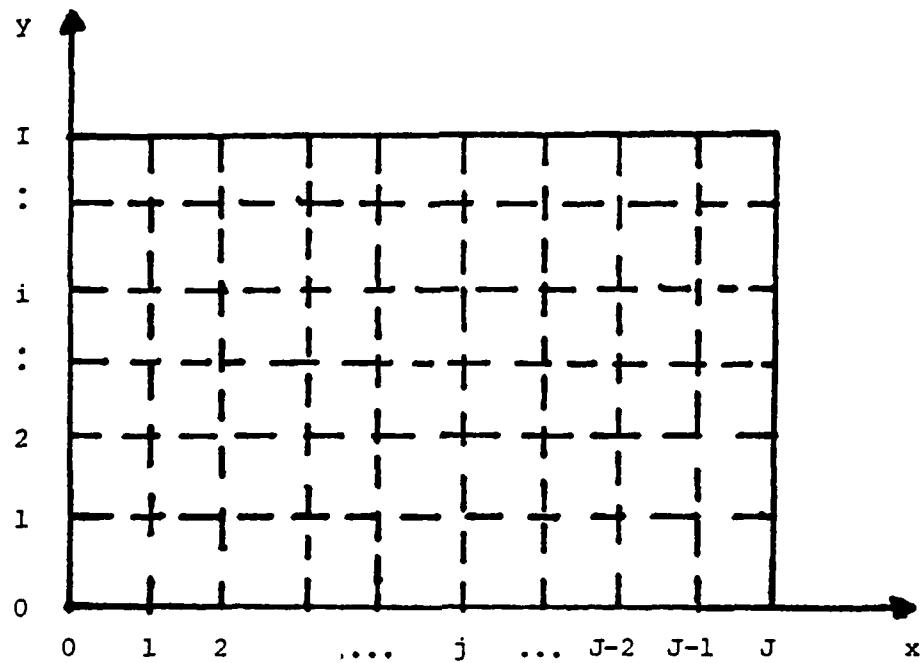


Figure 1. Grid Numbering Scheme for Boundary Condition Formulation

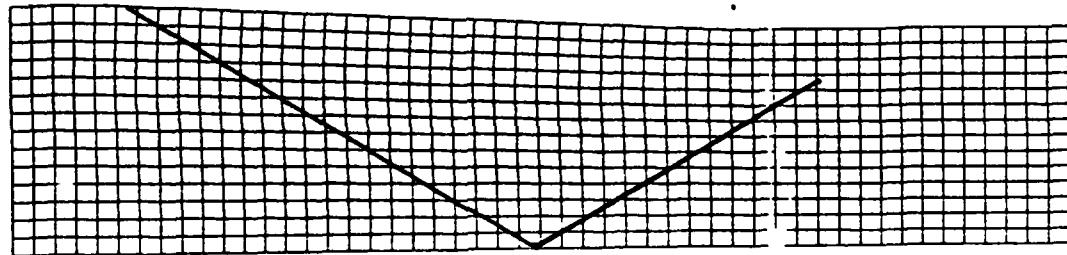


Figure 2. Computational Grid for Weak Shock Diffuser Calculations

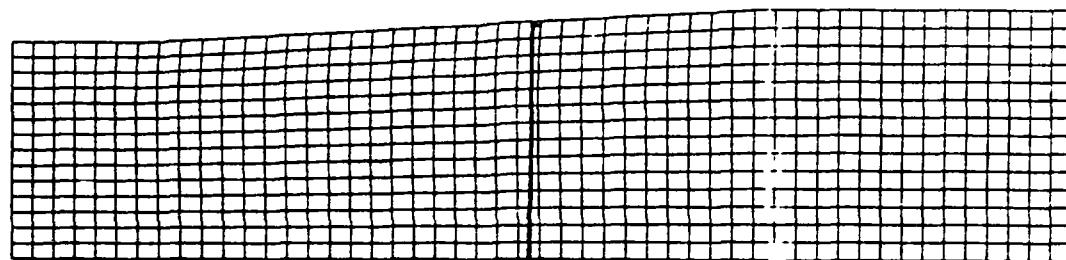


Figure 3. Computational Grid for Strong Shock Diffuser Calculations

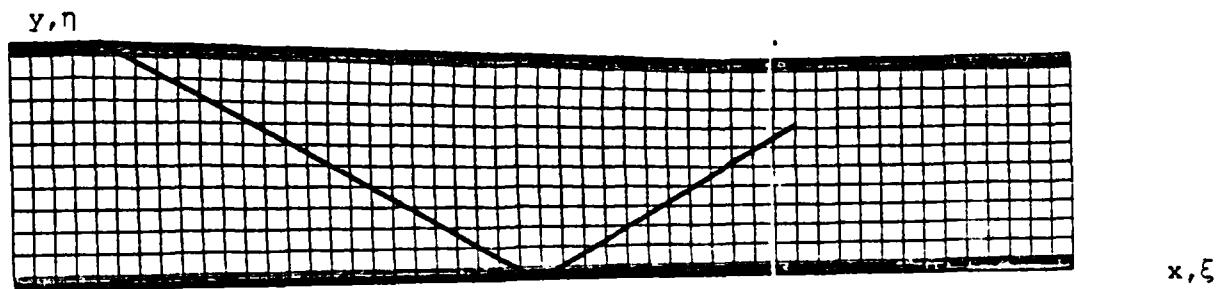


Figure 4. Computational Grid for Shock-Boundary Layer Computation

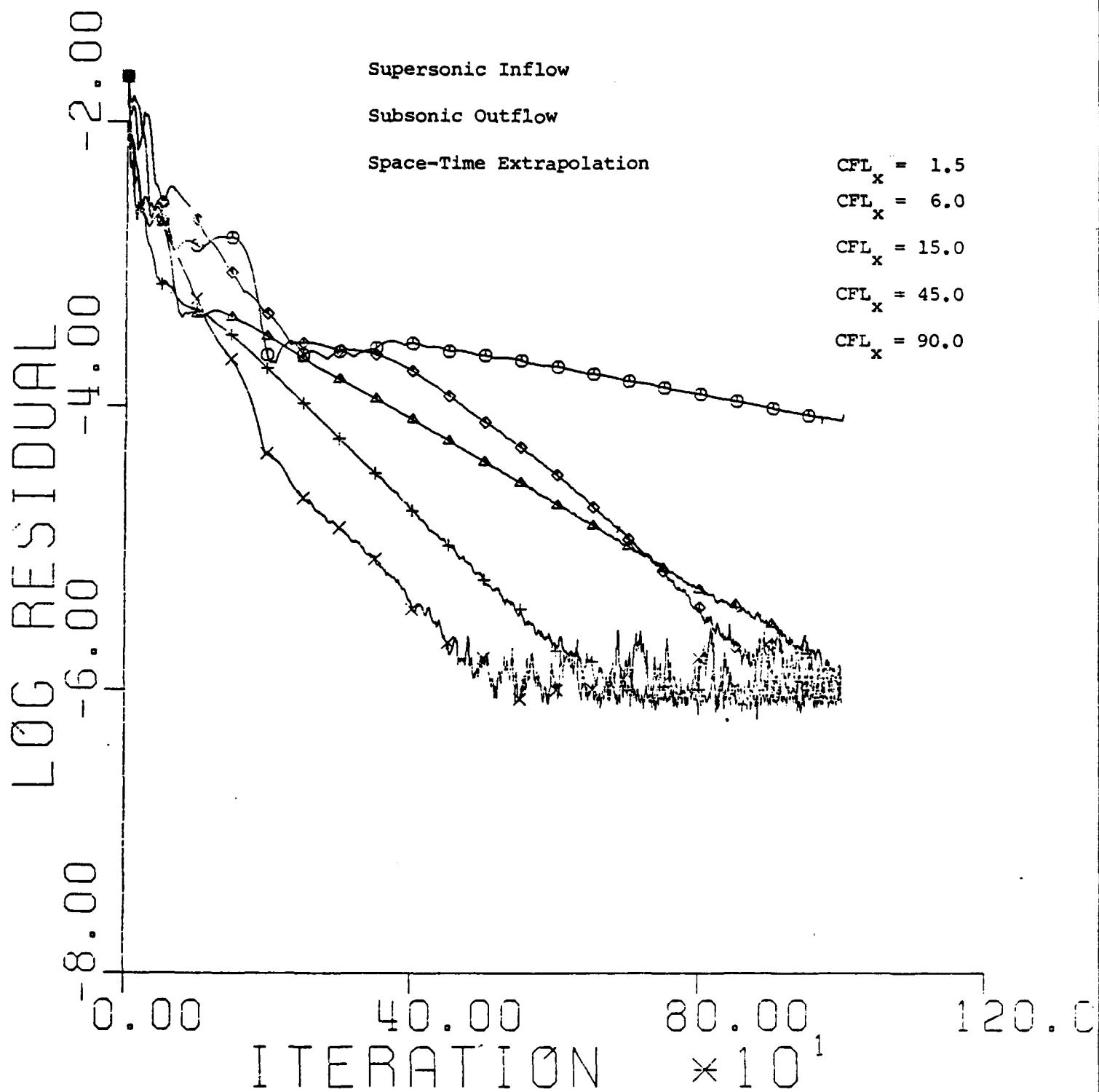


Figure 5. Convergence History for Strong Shock Diffuser Calculation

GRID GENERATION EXPERIENCE

Summary

One of the most difficult tasks to be performed as part of accurate computations for turbine blade geometries is grid generation. As part of our efforts in grid generation, we have constructed a coordinate generating scheme [BOGG] based on elliptic (Poisson) equation solutions which seem attractive for internal flow calculations. These coordinate systems are either orthogonal at boundaries or may have any specified angle, are periodic upstream and downstream of blade rows, and may have arbitrary, user-specified spacing near boundaries. This grid system does not correct the problem of highly sheared grids and does not provide adequate grid resolution in the far field. Analysis of the calculational results indicates that the present use of the strong conservational law Navier-Stokes form and central finite differencing is the problem rather than the grid systems. Investigations are continuing to determine if simple, sheared grid systems may be used in place of the more complex BOGG grids.

BOGG - Introduction

One of the major problems in developing an efficient numerical calculational scheme is the choice of a proper coordinate system, capable of transforming a complicated physical domain into a rectangular, evenly spaced calculational domain. In the viscous flow case, the coordinate system must give a high resolution region close to solid boundaries in order to capture the boundary layer, and its lines should be normal to this boundary. Especially demanding is the grid development for transonic compressor and turbine cascades due to their complicated shape, the complex flow structure and the presence of periodic boundaries. The practical problem usually

starts with a given grid point distribution along the boundaries coupled with some desired properties of the coordinate lines at the boundaries, and seeks a one-to-one mapping of the points from the physical domain into the computational domain. In recent studies, many new methods of numerical coordinate system generation have been developed ([1]-[7]). Most of these methods are based on the solution of Poisson equations with Dirichlet boundary conditions, giving unique and continuously differentiable transformation functions. Only few of them, however, provide a direct control of the line direction and spacing. Thomas [3] provides analytical expressions for the controlling functions, but this method fails to produce the desired results in all but the simplest cases. Sorenson and Steger [4] developed an effective method of controlling the spacing and angles at boundaries for O- and C-type grid systems. Warsi and Thompson [6] introduced a non-iterative method for the numerical generation of orthogonal curvilinear coordinates for plane annular regions between two smooth closed curves. Although most of these methods are adequate for some problems, they are not fully suitable for grid generation in internal flow calculations. The grid generation method for this type of problem has to provide a positive control of the grid lines at all boundaries, give the required spacing, generate coordinate lines without excessive skewness while being simple, fast and generally applicable.

BOGG uses a system of grid generating elliptic (Poisson) equations similar to those in [3], but it introduces a simple and effective iteration method of controlling the coordinate line angles at all boundaries. Coordinate systems were developed for different inviscid and viscous internal flow problems and used with success in an implicit two-dimensional numerical scheme [11].

BOGG Basic Formulation

The present method seeks a coordinate system transform

$$\xi = \xi(x, y) \quad \text{and} \quad \eta = \eta(x, y)$$

as a solution of the Poisson equations

$$\begin{aligned} \xi_{xx} + \xi_{yy} &= P(\xi, \eta) \\ \eta_{xx} + \eta_{yy} &= Q(\xi, \eta) \end{aligned} \tag{1}$$

where x, y are the Cartesian coordinates in the physical domain Ω , and

ξ, η are rectangular coordinates in the calculational domain (Fig. 1).

The coordinate lines $\xi = \text{const}$ were in most of the cases aligned streamwise and included solid surfaces; the coordinate lines $\eta = \text{const}$ were usually normal to the streamwise direction at the boundaries. Equation (1) is subject to the Dirichlet boundary conditions $\xi_b = \xi_b(x, y)$ and $\eta_b = \eta_b(x, y)$. The right-hand side functions P and Q control the spacing of the ξ and η lines inside the domain Ω . Since the values of ξ and η are fixed on the boundary, a change in spacing within Ω will produce a change of the angle at which the ξ and η lines intersect the boundary. The correct value of P and Q is not known a priori, so that an iterative procedure over the system (1) is necessary to find P and Q .

The effect of the functions $P(\xi, \eta)$ and $Q(\xi, \eta)$ on the resulting coordinate system can be seen by comparing the solution of (1) with the corresponding homogeneous system

$$\begin{aligned} \xi_{xx} + \xi_{yy} &= 0 \\ \eta_{xx} + \eta_{yy} &= 0 \end{aligned} \tag{2}$$

Let ξ, η be the solution of (1) and ξ^*, η^* the solution of (2), subject to the same boundary conditions as (1). Then, in a subregion D of Ω , a

negative $P(\xi_m, \eta)$ will cause a line $\xi = l = \text{const}$ to move closer to the line ξ_m than the line $\xi^* = l$, since ξ and η are subharmonic on D . Similarly, a negative $Q(\xi, \eta_m)$ will move a line $\eta = l = \text{const}$ closer to the line η_m than the line $\eta^* = l$. Positive values of P and Q have the opposite effect on the ξ and η lines. If the curve $\xi = \xi_m$ is now a branch cut in D , this branch cut will move in the direction of increasing ξ for a positive $P(\xi_m, \eta)$. The magnitude of P and Q determine the magnitude of the movement of the coordinate lines; the sign of P and Q determine the direction of this movement.

The situation at a boundary $\eta = \text{const} = \eta_{\min}$ is shown in Fig. 2a. Here the position of the $\xi = \text{const}$ line on the boundary is fixed by the boundary condition $\xi = \xi_b(x, y)$. A negative increment in $P(\xi, \eta_{\min})$ will cause the line ξ to move inside of Ω in the direction of increasing ξ . This movement will decrease the angle γ between the line $\xi = \text{const}$ and the tangent \vec{t} to the boundary. A positive increment in $P(\xi, \eta_{\min})$ will cause a movement of the ξ line in opposite direction. A proper value of P can therefore give the desired angle $\gamma = \beta$, $0 < \beta < \pi$. Here it should be remembered, however, that the influence of $P(\xi, \eta_{\min})$ is not limited to the line ξ but affects the coordinate lines in the entire region. In most of the practical cases the angle forcing requirement is not limited to one point but extends at least over some part of the boundary $\eta = \text{const}$. This means that $P(\xi, \eta)$ has to be found for all points simultaneously rather than by pointwise calculation.

Fig. 2b shows the analogous case of the influence of Q on $\eta = \text{const}$ lines at the boundary $\xi = \text{const} = \xi_{\max}$. Here a positive increment of $Q(\xi_{\max}, \eta)$ will decrease the angle γ .

For practical calculations the inverse relationship

$$\begin{aligned} x &= x(\xi, \eta) \\ y &= y(\xi, \eta) \end{aligned} \tag{3}$$

is needed. The inverse form of the system (1) is (see for example [1])

$$\begin{aligned} \alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} &= -J^2(Px_\xi + Qx_\eta) \\ \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} &= -J^2(Py_\xi + Qy_\eta) \end{aligned} \tag{4}$$

where

$$\begin{aligned} \alpha &= x_\eta^2 + y_\eta^2 \\ \beta &= x_\eta x_\xi + y_\eta y_\xi \\ \gamma &= x_\xi^2 + y_\xi^2 \end{aligned}$$

$J = x_\xi y_\eta - x_\eta y_\xi$ = Jacobian of the transform

It is of advantage to define new spacing controlling functions as

$$\phi = P \frac{J^2}{\alpha} \quad \psi = Q \frac{J^2}{\gamma} \tag{5}$$

The values of the functions ϕ and ψ are several orders of magnitude smaller than P and Q . The above explanation for the effect of P and Q on the grid lines applies to ϕ and ψ without restrictions since $J^2/\alpha > 0$ and $J^2/\gamma > 0$. With (5), the equations (4) become finally

$$\begin{aligned} \alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} &= (\alpha\phi x_\xi + \gamma\psi x_\eta) \\ \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} &= (\alpha\phi y_\xi + \gamma\psi y_\eta) \end{aligned} \tag{6}$$

subject to the boundary conditions $x = x_b(\xi, \eta)$, $y = y_b(\xi, \eta)$. Once ϕ and ψ are chosen, the system (6) can be solved using an appropriate numerical method.

Choice of ϕ and ψ

The previous discussion illustrates that the angle γ at the boundaries is determined by the function ϕ at $\eta = \text{constant}$ boundaries, and the function ψ at $\xi = \text{constant}$ boundaries. Starting from initial values of ϕ and ψ , the controlling functions have to be adjusted in an iterative manner to give the required angles (β) between the coordinate line and the boundary.

In general the left boundary is defined by $\xi = \xi_{\min}$, the right boundary by $\xi = \xi_{\max}$, and the lower and upper boundaries by $\eta = \eta_{\min}$ and $\eta = \eta_{\max}$ respectively. Solid walls are generally located on part or all of the range $\xi = \xi_{\min}$ to ξ_{\max} at $\eta = \eta_{\min}$ and $\eta = \eta_{\max}$. The most common requirement for this type of grid is that the coordinate lines are orthogonal to the boundaries ($\beta = \pi/2$, $\alpha = 0$). Specification of an angular deviation from orthogonality is, however, allowed (i.e. the angle α is specified non-zero). The following discussion applies both to orthogonal and non-orthogonal grid line intersections at the boundaries.

The procedure necessary to obtain the desired grid angle, β , at the lower boundary will now be discussed. The nomenclature used in the following section is as defined in Fig. 2a. It is assumed in this correction procedure that any change in the forcing function ϕ will have only a localized effect on the coordinate lines.

It was stated in the previous discussion that decrementing the forcing function ϕ (i.e. P) will reduce the angle γ . Similarly incrementing ϕ will increase the angle γ . The following conditions therefore apply.

$$\begin{aligned} \gamma > \beta & \text{ we require } \Delta\phi < 0 \\ \gamma < \beta & \text{ we require } \Delta\phi > 0 \end{aligned} \tag{7}$$

Choice of an incremental function to satisfy these criteria will force the grid lines to the required orientation. The forcing function ϕ is modified between iteration counts l and $l+1$ using the equation

$$\phi(\xi, \eta_{\min})^{l+1} = \phi(\xi, \eta_{\min})^l + \Delta\phi^l \quad (8)$$

The correction factor, $\Delta\phi^l$, used in this computation is given by

$$\Delta\phi^l = \frac{-\sin(\gamma - \beta)}{A} \quad (A > 1) \quad (9)$$

This expression satisfies the movement criteria expressed in (7). The damping factor $A (> 1.0)$ is introduced to control iteration stability.

A similar analysis at the upper boundary leads to the following iteration sequence:

$$\phi(\xi, \eta_{\max})^{l+1} = \phi(\xi, \eta_{\max})^l + \frac{\sin(\gamma - \beta)}{A} \quad (10)$$

The value of ϕ inside the domain Ω is obtained by linear interpolation along a line $\xi = \text{constant}$.

$$\phi(\xi, \eta) = \phi(\xi, \eta_{\min}) + \left(\phi(\xi, \eta_{\max}) - \phi(\xi, \eta_{\min}) \right) \left(\frac{\eta - \eta_{\min}}{\eta_{\max} - \eta_{\min}} \right) \quad (11)$$

At the boundary $\xi = \xi_{\max}$, ψ is determined by the sequence

$$\psi(\xi_{\max}, \eta)^{l+1} = \psi(\xi_{\max}, \eta)^l + \frac{\sin(\gamma - \beta)}{B} \quad . \quad (12)$$

and at $\xi = \xi_{\min}$.

$$\psi(\xi_{\min}, \eta)^{l+1} = \psi(\xi_{\min}, \eta)^l - \frac{\sin(\gamma - \beta)}{B} \quad (13)$$

The angles used in equation (12) are defined in Fig. 2b. Between these boundaries ψ is calculated along a line $\eta = \text{const}$ by linear interpolation.

$$\psi(\xi, \eta) = \psi(\xi_{\min}, \eta) + \left(\psi(\xi_{\max}, \eta) - \psi(\xi_{\min}, \eta) \right) \left(\frac{\xi - \xi_{\min}}{\xi_{\max} - \xi_{\min}} \right) \quad (14)$$

If no specific requirement is enforced on the grid intersection angles at any boundary, the forcing function remains constant throughout the iteration procedure.

The above correction procedures require evaluation of the sine of the angular error. This sine can be expanded (on the lower and upper boundaries) to:

$$\begin{aligned}\sin(\gamma-\beta) &= \sin[(\gamma+\theta) - (\beta+\theta)] \\ &= \sin(\gamma+\theta) \cos(\beta+\theta) - \cos(\gamma+\theta) \sin(\beta+\theta)\end{aligned}\quad (15)$$

where the angles are defined in Fig. 2a for the lower boundary. An angular deviation (α) from orthogonality may be specified so the above equation is rewritten in terms of α , γ and θ as:

$$\sin(\gamma-\beta) = -\sin(\gamma+\theta) \sin(\theta-\alpha) - \cos(\gamma+\theta) \cos(\theta-\alpha)$$

The angle, $\theta-\alpha$, is constant throughout the iteration procedure and is therefore only calculated once. The sine and cosine of the angle between the $\xi = \text{constant}$ line and the x -axis ($\gamma+\theta$) are evaluated using the equations:

$$\begin{aligned}\sin(\gamma+\theta) &= \frac{y(\xi, \eta_p) - y(\xi, \eta_{p-1})}{\sqrt{\left(x(\xi, \eta_p) - x(\xi, \eta_{p-1})\right)^2 + \left(y(\xi, \eta_p) - y(\xi, \eta_{p-1})\right)^2}} \\ \cos(\gamma+\theta) &= \frac{x(\xi, \eta_p) - x(\xi, \eta_{p-1})}{\sqrt{\left(x(\xi, \eta_p) - x(\xi, \eta_{p-1})\right)^2 + \left(y(\xi, \eta_p) - y(\xi, \eta_{p-1})\right)^2}}\end{aligned}\quad (16)$$

where

$$\eta_p = \eta_{\min+1} \text{ on the lower boundary}$$

$$\eta_p = \eta_{\max} \text{ on the upper boundary}$$

On the left and right boundaries, the sine of the angular error is expressed as:

$$\begin{aligned}
 \sin(\gamma-\beta) &= \sin[(\gamma-\theta) - (\beta-\theta)] \\
 &= \sin(\gamma-\theta) \cos(\beta-\theta) - \cos(\gamma-\theta) \sin(\beta-\theta) \\
 &= \sin(\gamma-\theta) \sin(\theta+\alpha) - \cos(\gamma-\theta) \cos(\theta+\alpha)
 \end{aligned} \tag{17}$$

The angle, $\theta+\alpha$, is again constant throughout the iteration procedure, and the sine and cosine of the angle between the $\eta = \text{constant}$ line and the x -axis ($\gamma-\theta$) are evaluated using the equations

$$\sin(\gamma-\theta) = \frac{-(y(\xi_p, \eta) - y(\xi_{p-1}, \eta))}{\sqrt{(x(\xi_p, \eta) - x(\xi_{p-1}, \eta))^2 + (y(\xi_p, \eta) - y(\xi_{p-1}, \eta))^2}} \tag{18}$$

$$\cos(\gamma-\theta) = \frac{x(\xi_p, \eta) - x(\xi_{p-1}, \eta)}{\sqrt{(x(\xi_p, \eta) - x(\xi_{p-1}, \eta))^2 + (y(\xi_p, \eta) - y(\xi_{p-1}, \eta))^2}} \tag{19}$$

where

$$\xi_p = \xi_{\min+1} \text{ on the left boundary}$$

$$\xi_p = \xi_{\max} \text{ on the right boundary}$$

BOGG Iteration Procedures

Once the functions ϕ and ψ are established the system of Poisson equations (6) can be numerically solved using finite difference methods. The scheme selected was successive line over relaxation (SLOR). SLOR is first applied along ξ lines and then η lines for a number of iterations. The forcing functions are then corrected and the procedure repeated.

The over-relaxation factors used in the SLOR calculations were generally taken as 1.15 for evaluation along ξ lines and 1.05 along η lines. The damping factors applied to the forcing function corrections [Eqs. (9), (10), (12), and (13)] were generally the same and had values in the range 1 to 5.

BOGG Calculation Examples

The simplest type of coordinate system was that of the diffuser, which consists of an upstream boundary $\xi = 0$, downstream boundary $\xi = i_{\max}$, solid wall boundary at $\eta = 0$ and either solid wall or symmetry line at $\eta = j_{\max}$. The lines $\xi = \text{const}$ were required to be normal to the η boundaries; there was no specific requirement for the $\eta = \text{const}$ lines at the ξ boundaries in the inviscid case. The viscous grid has geometric point distribution along the ξ lines allowing proper boundary layer resolution. The spacing in the ξ direction is constant. The maximum error criteria for θ was 0.2° . The resulting grid systems can be seen in Figs. 3, 4 and 5.

Compressor cascades are difficult geometries for developing coordinate systems suitable for efficient viscous and non-viscous calculations. The blade shapes chosen for examples were designed by Tong [10] with help of an inverse numerical code. The physical domain consisted of the region between the two blades and an upstream and downstream region bounded by blade chord extension lines. Along these lines, the periodic boundary conditions have to be applied, requiring the same point distribution at the upper and lower boundary upstream and downstream of the blade, respectively. The upstream ($\xi = 0$) and downstream ($\xi = 40$) boundaries were located one chord length from the leading and trailing edge, respectively. The viscous grid has the first and last η lines densely packed (geometric distribution) in order to capture the boundary layer; the remaining η lines are uniformly spaced. In the non-viscous grid all the η lines are uniformly spaced. The ξ lines ahead and behind the blade have geometric spacing to provide higher resolution close to the leading and trailing edge. The maximum error of θ was 0.5° . The resulting grids are shown in Figs. 6 and 7. These grids were used with

considerable success by Bush [9, 11] in his two-dimensional viscous implicit code.

The most demanding example was to generate a grid for finite difference calculations of the flow properties in a turbine cascade. The transonic turbine blade shape was the same as used by Demuren [12]. The $\xi = 0$ line is the U-shaped upstream and periodic boundary extending from the leading edge of the lower blade to one chord length ahead of the blade to the leading edge of the upper blade. The periodic boundary is located at the first and last 8 points on the line $\xi = 0$; between these two regions is the upstream boundary. The downstream boundary corresponds to $\xi = 40$.

The inviscid grid in Fig. 8 has $30\eta = \text{constant}$ lines. The viscous grid has $60\eta = \text{constant}$ lines, where the first and last 20 points have geometric distribution that allows good resolution of the boundary layer, see Fig. 9. The requirement for the ξ -lines was again to be normal to the $\eta = 0$ and $\eta = j_{\max}$ boundaries. The resulting C-type grids are shown in Figs. 8 and 9.

The versatility of the present method can be best demonstrated on the example of grid generation in the axial-radial plane of a transonic compressor with both rotor and stator. The grid consists of five different regions (upstream of rotor, rotor, between rotor and stator, stator and downstream of stator) that are treated separately and then joined together. In this case, both line types $\xi = \text{constant}$ and $\eta = \text{constant}$ are controlled at the corresponding boundaries. The angle between the $\xi = \text{constant}$ line and the upper and lower boundary changes gradually from $\pi/2$ to the angle between $\xi = i_{\max}$ line and either the line $\eta = 0$ or $\eta = j_{\max}$. The $\eta = \text{constant}$ lines are manipulated at the $\xi = 0$ and $\xi = i_{\max}$ boundaries to give smooth transition

between two adjoining regions. The ξ = constant lines have geometric distribution near the leading and trailing edges. There are altogether 98 ξ -lines and 17 η -lines. The resulting coordinate system is shown in Fig. 10.

BOGG Input Details

BOGG requires two data files for input. One contains the coordinates of the grid boundary, the other switches and factors required during the calculation procedure.

The input files have the following format:

X-Y coordinate file

X,Y Values on the lower surface - format [2X,(8E14 .7)]

X,Y Values on the upper surface

X,Y Values on the left surface

X,Y Values on the right surface

Setup file

Line 1: Number of η lines, number of ξ lines - format (2X,2I5)

Line 2: Number of forcing function correction iterations, number
of SLOR iterations - format (2X,2I5)

Line 3: Damping factors A and B - format (2X,2f10 .6)

Line 4: Over-relaxation factors applied along ξ lines and η lines -
format (2X,2f10 .6)

Line 5: Minimum angle deviation for convergence (RAD) - format
(2X,f10 .6)

Line 6: Intersection angle options on lower, upper, left and right
boundaries ISTART, IEND, IANG - Format (2X,3I5)
where ISTART and IEND are the positions of the start and

end of the angle option and

IANG = 0 denotes intersection angle not forced

IANG = 1 denotes grid line orthogonal to boundary

IANG = 2 denotes angular deviation from orthogonality specified.

If IANG equals 2, the angular deviation (RAD) is then

input - format [2X,(8E14.7)].

BOGG Output Details

The program generates an X-Y coordinate file of grid intersection locations. At each ξ = constant line (moving from ξ_{\min} to ξ_{\max}), the X coordinates are output for $\eta = \eta_{\min}$ to $\eta = \eta_{\max}$ [format(2x(8E14.7))], followed by the Y coordinates in the same order.

Geometry Interpolation Codes

A series of computer codes has been written as an interface between Rolls-Royce turbine geometric description and standard GTL usage [13]. A pre-processor reads a Rolls-Royce proconsul file of blade coordinates and generates blade profile coordinates in a form compatible with standard GTL usage. Two more codes interpolate this data, using a Rolls-Royce supplied spline interpolation routine, on to user specified X.cuts and interface with the grid generation programs. A fourth program has been written to interactively display and revise the resulting geometries and grid lines.

Grid-Related Errors

As a test case for turbine type geometries, an inviscid calculation of the flow through a turbine cascade was attempted.

The grid chosen was a throughflow type grid as shown in Figure 11. This grid had 40 points in the axial direction and 20 points in the tangential direction. Calculations based on this grid were extremely

disappointing especially in terms of stagnation pressure error. A contour plot of stagnation pressure loss for this turbine cascade is shown in Figure 12. upstream of shock waves, any stagnation pressure loss is to be considered a solution error. The maximum error is about 15% near the leading edge, but a 10% error is observed in the inlet region where the solution should and does have very nearly uniform velocity.

Since this type of stagnation pressure error is known to be common to most time marching algorithms, considerable time was devoted to analyzing the error mechanism for Beam and Warming type codes. It was felt that there were two distinct error mechanisms, one associated with solid wall boundary conditions and a second one associated with the truncation error of the scheme. It was found that the stagnation pressure errors upstream of the blade rows correlated well with the second mechanism and were not due to contamination of the solution by boundary condition errors.

In order to understand the truncation error of the steady state finite difference solutions, it is necessary to examine the strong conservation law form of the equations being solved which are the two dimensional, transformed, Euler equation:

$$\bar{U}_t + \bar{E}_\xi + \bar{F}_\eta = 0 \quad (20)$$

$$\bar{U} = J^{-1} \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix} \quad \begin{aligned} \bar{E} &= (\xi_x^E + \xi_y^F) J^{-1} \\ \bar{F} &= (\eta_x^E + \eta_y^F) J^{-1} \end{aligned} \quad (21)$$

$$\mathbf{E} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho h_o u \end{pmatrix} \quad \mathbf{F} = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho h_o v \end{pmatrix} \quad (22)$$

$$J^{-1} = x_{\xi} y_{\eta} - x_{\eta} y_{\xi} \quad ; \quad J = 1/J^{-1} \quad (23)$$

$$\xi_x = J y_{\eta} \quad \xi_y = -J x_{\eta} \quad (24)$$

$$\eta_x = J y_{\xi} \quad \eta_y = J x_{\xi}$$

The values of the metric quantities (J^{-1} , ξ_x , ξ_y , η_x , η_y) are evaluated numerically using the grid node coordinates. The steady state finite difference form of equation (20) becomes:

$$\frac{\bar{E}_{j+1,k} - \bar{E}_{j-1,k}}{2\Delta\xi} + \frac{\bar{F}_{j,k+1} - \bar{F}_{j,k-1}}{2\Delta\eta} = 0 \quad (25A)$$

or in operator notation

$$\frac{\delta_j \bar{E}}{2\Delta\xi} + \frac{\delta_k \bar{F}}{2\Delta\eta} = 0 \quad (25B)$$

In physical space variables, the continuity equation becomes:

$$\begin{aligned} & \frac{(\rho u)_{j+1,k}}{\Delta\xi} \frac{(y_{j+1,k+1} - y_{j+1,k-1})}{\Delta\eta} - \frac{(\rho u)_{j-1,k}}{\Delta\xi} \frac{(y_{j-1,k+1} - y_{j-1,k-1})}{\Delta\eta} \\ & - \frac{(\rho u)_{j,k+1}}{\Delta\eta} \frac{(y_{j+1,k+1} - y_{j-1,k+1})}{\Delta\xi} + \frac{(\rho u)_{j,k-1}}{\Delta\eta} \frac{(y_{j+1,k-1} - y_{j-1,k-1})}{\Delta\xi} \\ & - \frac{(\rho v)_{j+1,k}}{\Delta\xi} \frac{(x_{j+1,k+1} - x_{j+1,k-1})}{\Delta\eta} + \frac{(\rho v)_{j-1,k}}{\Delta\xi} \frac{(x_{j-1,k+1} - x_{j-1,k-1})}{\Delta\eta} \\ & + \frac{(\rho v)_{j,k+1}}{\Delta\eta} \frac{(x_{j+1,k+1} - x_{j-1,k+1})}{\Delta\xi} - \frac{(\rho v)_{j,k-1}}{\Delta\eta} \frac{(x_{j+1,k-1} - x_{j-1,k-1})}{\Delta\xi} = 0 \end{aligned} \quad (26A)$$

as illustrated in figures 13 and 14.

In operator notation we have

$$\begin{aligned} & (\rho u)_{j+1} \delta_k y_{j+1} - (\rho u)_{j-1} \delta_k y_{j-1} + (\rho u)_{k-1} \delta_j y_{k-1} - (\rho u)_{k+1} \delta_j y_{k+1} \\ & + (\rho v)_{j-1} \delta_k x_{j-1} - (\rho v)_{j+1} \delta_k x_{j+1} + (\rho v)_{k+1} \delta_j x_{k+1} - (\rho v)_{k-1} \delta_j x_{k-1} = 0 \end{aligned} \quad (26B)$$

This flux balance has the correct cell face area associated with the appropriate flux terms, but the flux across any face is approximated as the value at the face mid-node $(\rho u)_{j+1} \delta_k y_{j+1}$, for example. This flux balance is common to all schemes using the strong conservation law form and 2nd order central differencing.

One estimate of the error in this flux balance can be obtained if one estimates the same flux balance on a grid twice as fine as the present grid, as illustrated in Figure 15. The new flux sum will be:

$$\frac{\bar{E}_{j+\frac{1}{2}, k} - \bar{E}_{j-\frac{1}{2}, k}}{\Delta\xi} + \frac{\bar{F}_{j, k+\frac{1}{2}} - \bar{F}_{j, k-\frac{1}{2}}}{\Delta\eta} = 0 \quad (27)$$

We will estimate the term, $\bar{E}_{j+\frac{1}{2}, k}$, as:

$$\begin{aligned} \bar{E}_{j+\frac{1}{2}, k} &= \left(E \frac{\xi_x}{J} + F \frac{\xi_y}{J} \right)_{j+\frac{1}{2}, k} \\ &= \frac{1}{4} \left((E_{j+1, k} + E_{j, k}) (\delta_k y_{j+1} + \delta_k y_j) - (F_{j+1, k} + F_{j, k}) (\delta_k x_{j+1} + \delta_k x_j) \right) \end{aligned} \quad (28)$$

If we subtract equation (26B) from equation (28) we get a vector error estimate of:

$$\begin{aligned} \bar{\epsilon} &= (\delta_k y_j) (\delta_j E_k) + (\delta_j y_k) (\delta_k E_j) \\ &+ (\delta_j x_k) (\delta_k F_j) + (\delta_k x_j) (\delta_j F_k) \end{aligned} \quad (29)$$

Note that $\bar{\epsilon}$ is a vector, not a scalar quantity. Figure 16 shows a contour plot of ϵ_4 , error for energy equation, for the turbine cascade. The correspondence between this error parameter and the stagnation pressure error shown in Figure 12 is encouraging.

An unexpected result on this analysis is that one term in equation (29) is zero for sheared grids like those shown in Figure 17. For this grid, all y running coordinate lines are parallel, and $\hat{Q}_k x_j = 0$. This fact suggests that a sheared grid might reduce the stagnation pressure error and Figure 18 demonstrates that indeed it does.

Conclusions

The status of the grid generation effort is at present very open-ended. The grid error analysis shows that the scheme truncation error can be made as small as desired when the grid spacing is chosen small. In addition, one possible interpretation of the error analysis is that sheared grids will be adequate even for turbine type geometries. Considerable effort is now being devoted to determining if blunt leading and trailing edges and thin shear layers can be satisfactorily analyzed with these grids. Until these sheared grids can be shown to be inadequate, most computations will be done with the sheared grids rather than the BOGG grids. The BOGG grids do offer good leading edge resolution, but they do not solve the problem of sheared grid lines, see Figure 11 trailing edge region. In addition the far-field structure of these grids also appears to be a source for stagnation pressure errors.

Analysis of the present Beam and Warming type algorithmic use of the strong conservation law forms shows that the present algorithm flux balance is not very accurate. A primary goal for future code development is to improve this flux balance. Three possible methods to improve the flux balance are: one, introduce fourth order accurate finite difference forms on the steady state solution evaluation; second, introduce an improved, specialized flux balance like the error analysis in the steady state solution evaluation;

and third, introduce a combined finite difference - finite element formulation to provide the most accurate solutions. Each of these schemes will be investigated to determine their possible application.

REFERENCES

- 1 Thompson, J.F., Thames, F.C. and Mastin, C.W., "Boundary-Fitted Curvilinear Coordinate Systems for Solution of Partial Differential Equations on Fields Containing Any Number of Arbitrary Two-Dimensional Bodies." NASA CR-2729, July 1977.
- 2 Thompson, J.F., Thames, F.C. and Mastin, C.W., "TOMCAT--A Code for Numerical Generation of Boundary Fitted Curvilinear Coordinate Systems on Fields Containing Any Number of Arbitrary Two-Dimensional Bodies." J. of Computational Physics, Vol. 24, 1977, pp 274-302.
- 3 Thomas, P.D. and Middlecoff, J.F., "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations," AIAA Journal, Vol. 18, June 1980, pp 652-656.
- 4 Sorenson, R.L. and Steger, J.L., "Numerical Generation of Two-Dimensional Grids by the Use of Poisson Equations with Grid Control at Boundaries." Paper presented at NASA Workshop on Numerical Grid Generation, Langley Research Center, Hampton, VA, October 6-7, 1980.
- 5 Thompson, J.F. and Mastin, C.W., "Grid Generation Using Differential Systems Techniques." Paper presented at NASA Workshop on Numerical Grid Generation, Langley Research Center, Hampton, VA, October 6-7, 1980.
- 6 Warsi, Z.U.A. and Thompson, J.F., "Numerical Generation of Two-Dimensional Orthogonal Curvilinear Coordinates in a Euclidean Space." Paper presented at NASA Workshop on Numerical Grid Generation, Langley Research Center, Hampton, VA, October 6-7, 1980.
- 7 Steger, J.L. and Sorenson, R.L., "Use of Hyperbolic Partial Differential Equations to Generate Body Fitted Coordinates." Paper presented at NASA Workshop on Numerical Grid Generation, Langley Research Center, Hampton, VA, October 6-7, 1980.
- 8 Ives, D.C., "A Modern Look at Conformal Mapping, Including Doubly Connected Regions." AIAA Paper 75-842, AIAA 8th Fluid and Plasma Dynamics Conference, Hartford, CT 1975.
- 9 Bush, R.H., private communication
- 10 Thompkins, W.T., Jr. and Tong, S.S., "Inverse or Design Calculations for Non-Potential Flow in Turbomachinery Blade Passages." ASME Paper No. 81-GT-78, 1981.
- 11 Bush, R.H., "Time Accurate Internal Flow Solutions of the Thin Shear Layer Equations." MIT Gas Turbine Laboratory Report No. 156, Feb. 1981.
- 12 Demuren, H.O., "Aerodynamic Performance and Heat Transfer Characteristics of High Pressure Ratio Transonic Turbines." PhD Thesis, MIT, February 1976.

13 Norton, R.J.G., "Users Guide and Test Cases for the Pre- and Post-
Processors of the Grid Generation Program," Rolls-Royce, Inc. internal
memo, November 1981.

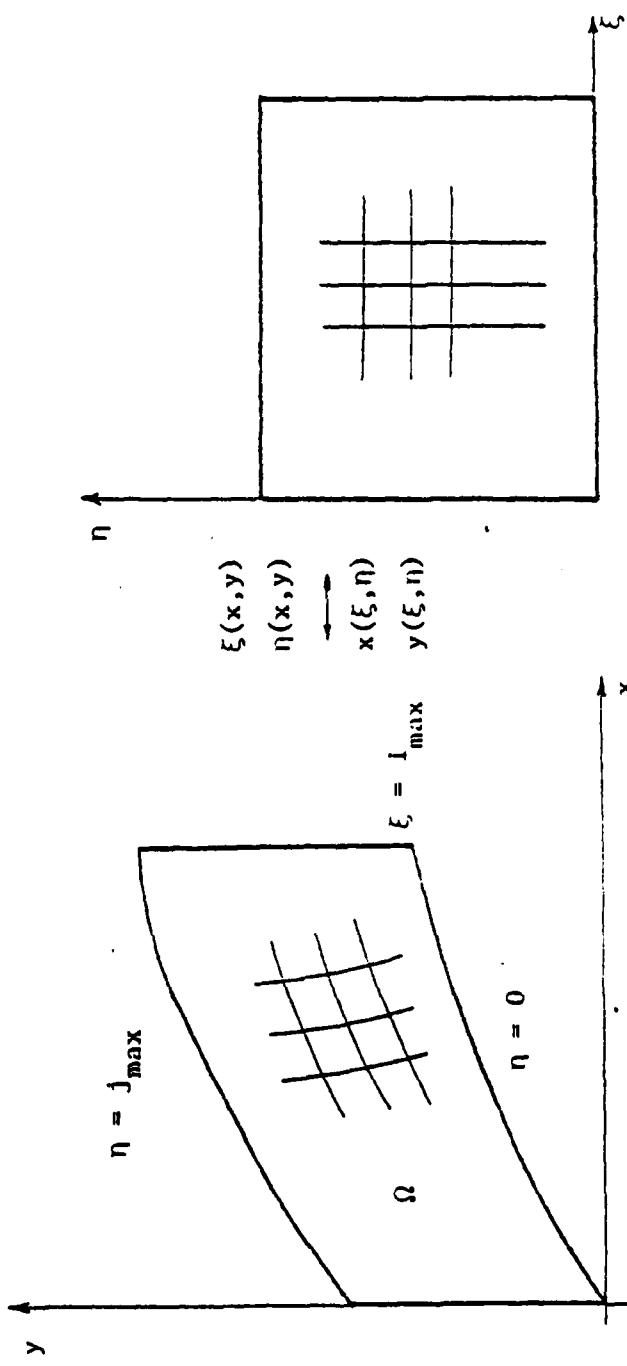
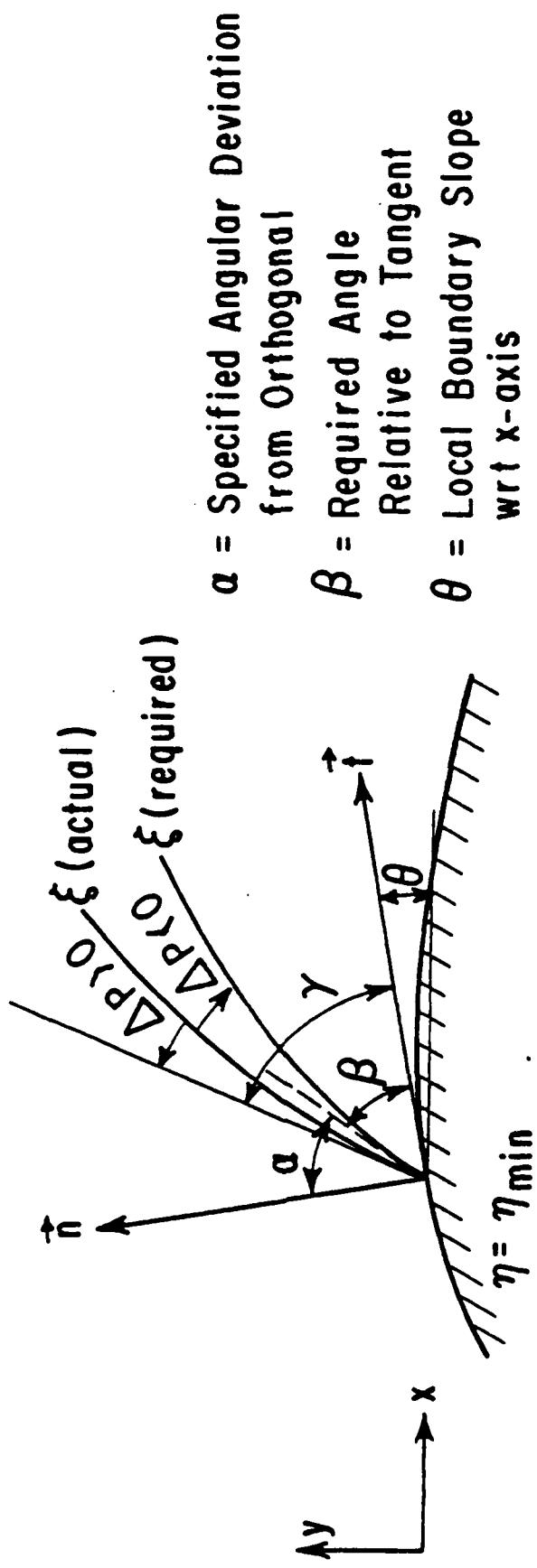


Fig. 1 Transforming physical domain Ω into computational domain.

EFFECT OF P ON ξ LINES AT THE BOUNDARY $\eta = \eta_{\min}$



α = Specified Angular Deviation
from Orthogonal

β = Required Angle
Relative to Tangent

θ = Local Boundary Slope
wrt x-axis

Figure 2a

EFFECT OF Q ON η LINES AT THE BOUNDARY $\xi = \xi_{\max}$

68

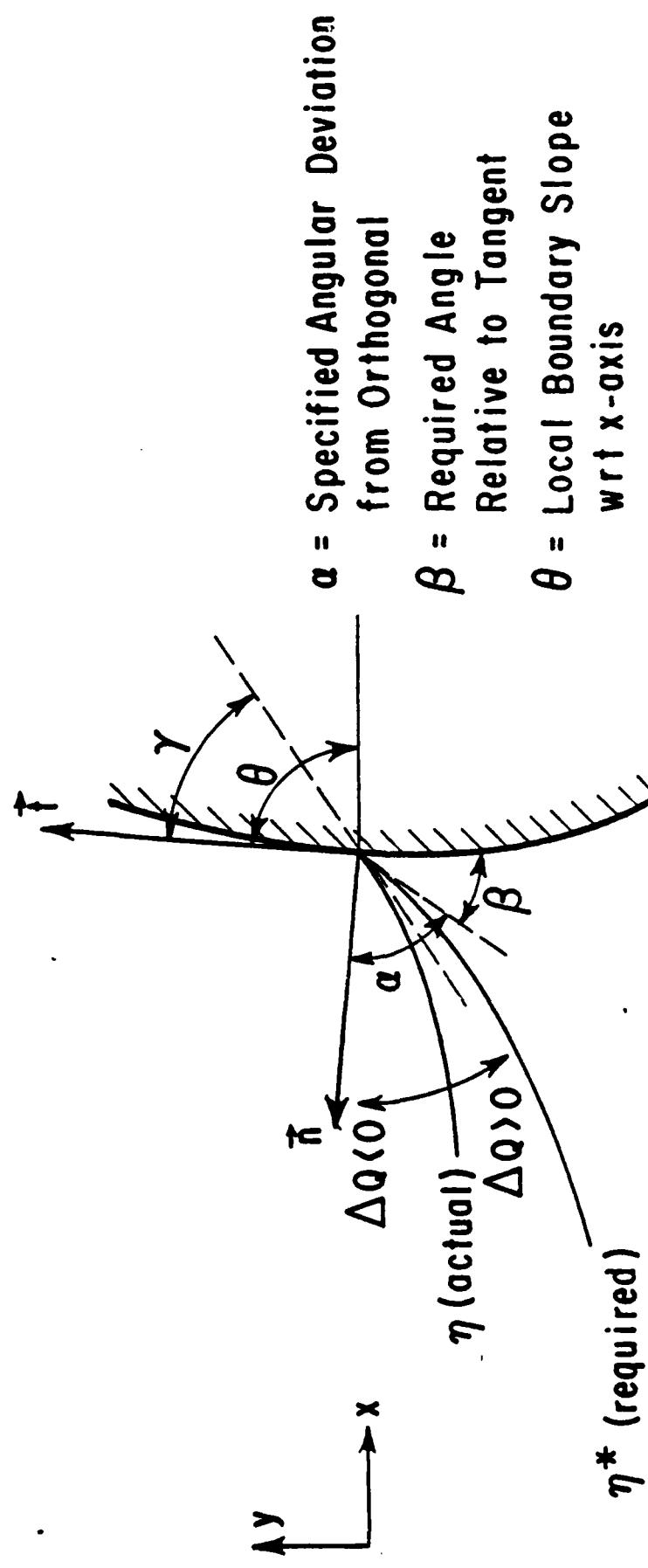


Figure 2b

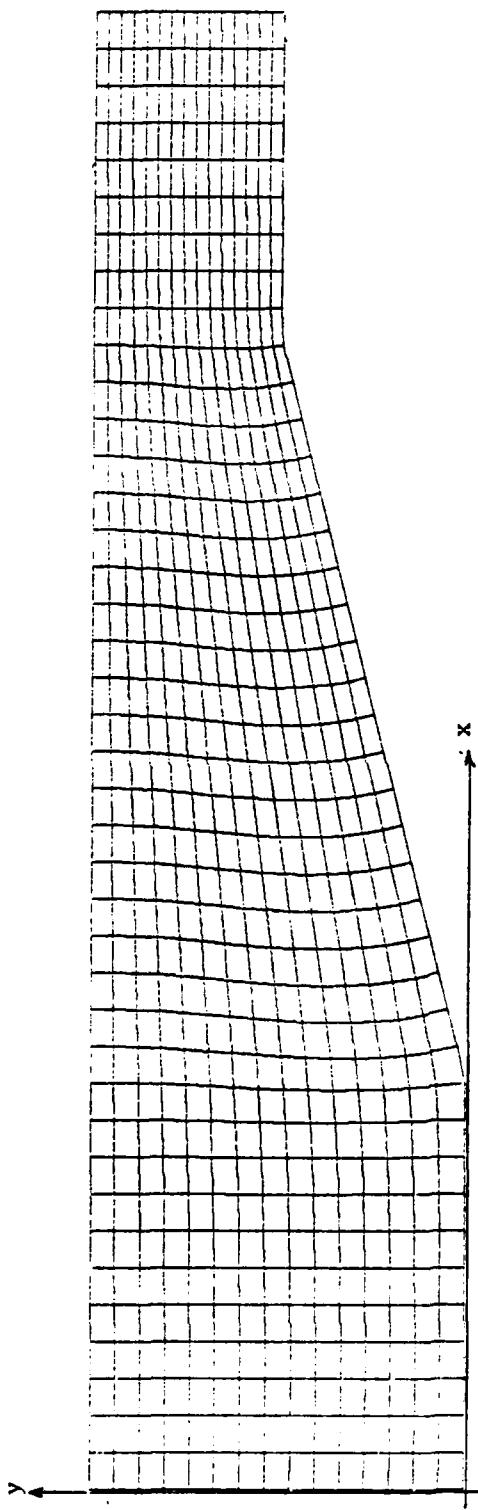


Fig. 3 Non-viscous diffusor grid,

$$\eta_{\max} = j_{\max} = 15,$$

$$\xi_{\max} = i_{\max} = 40$$

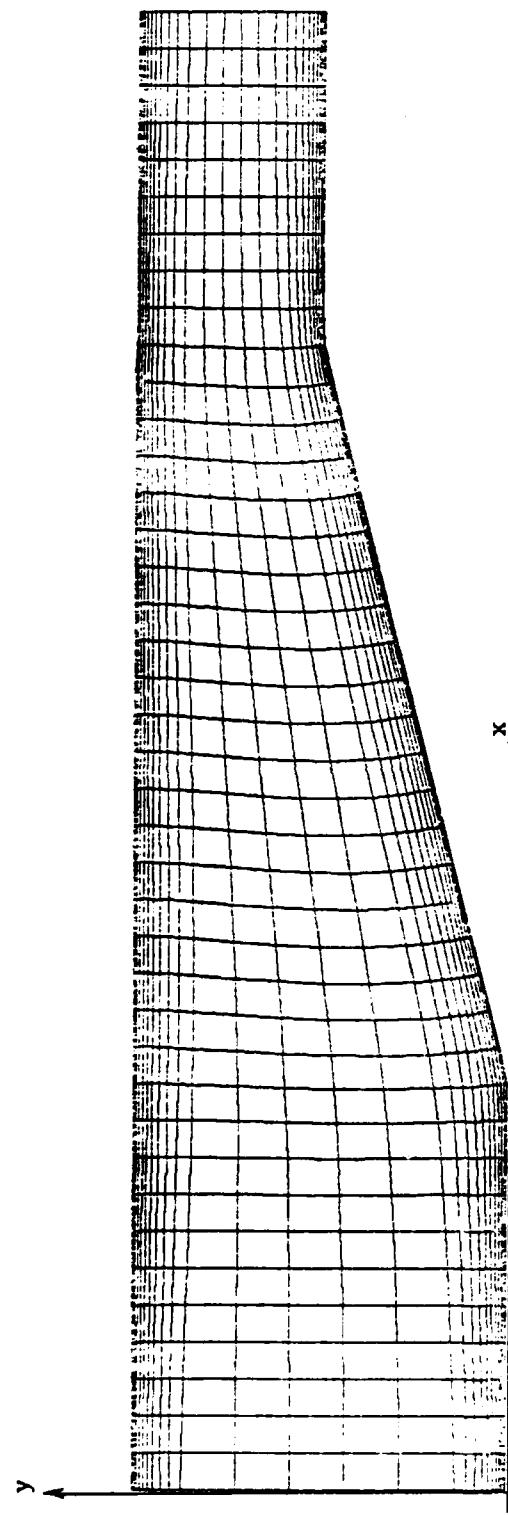


Fig. 4 Viscous diffusor grid;

$$\eta_{\max} = j_{\max} = 25$$

$$\xi_{\max} = i_{\max} = 40$$

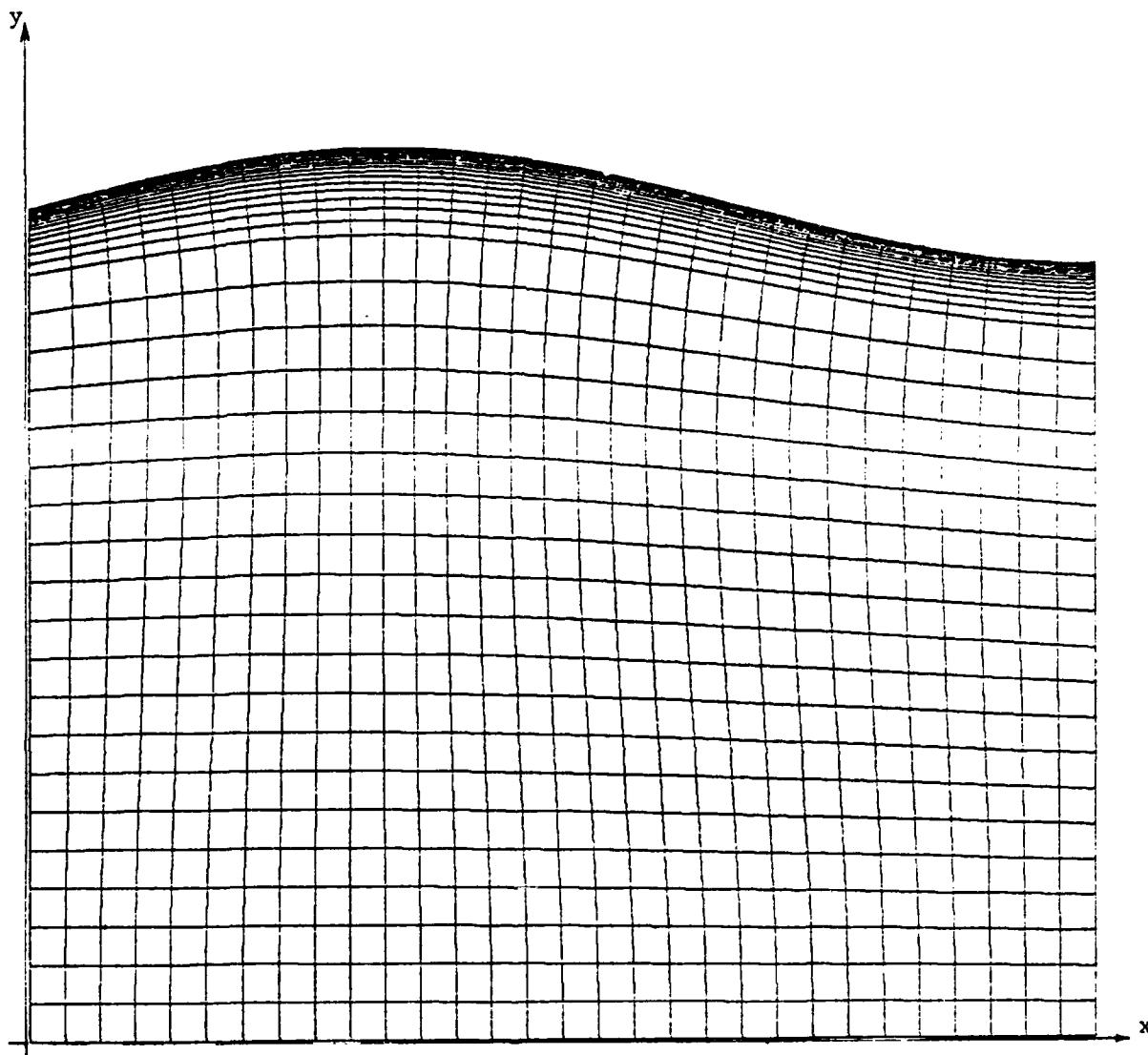


Fig. 5 Viscous diffusor grid with center line at $\eta = 0$

$$\eta_{\max} = j_{\max} = 40$$

$$\xi_{\max} = i_{\max} = 30$$

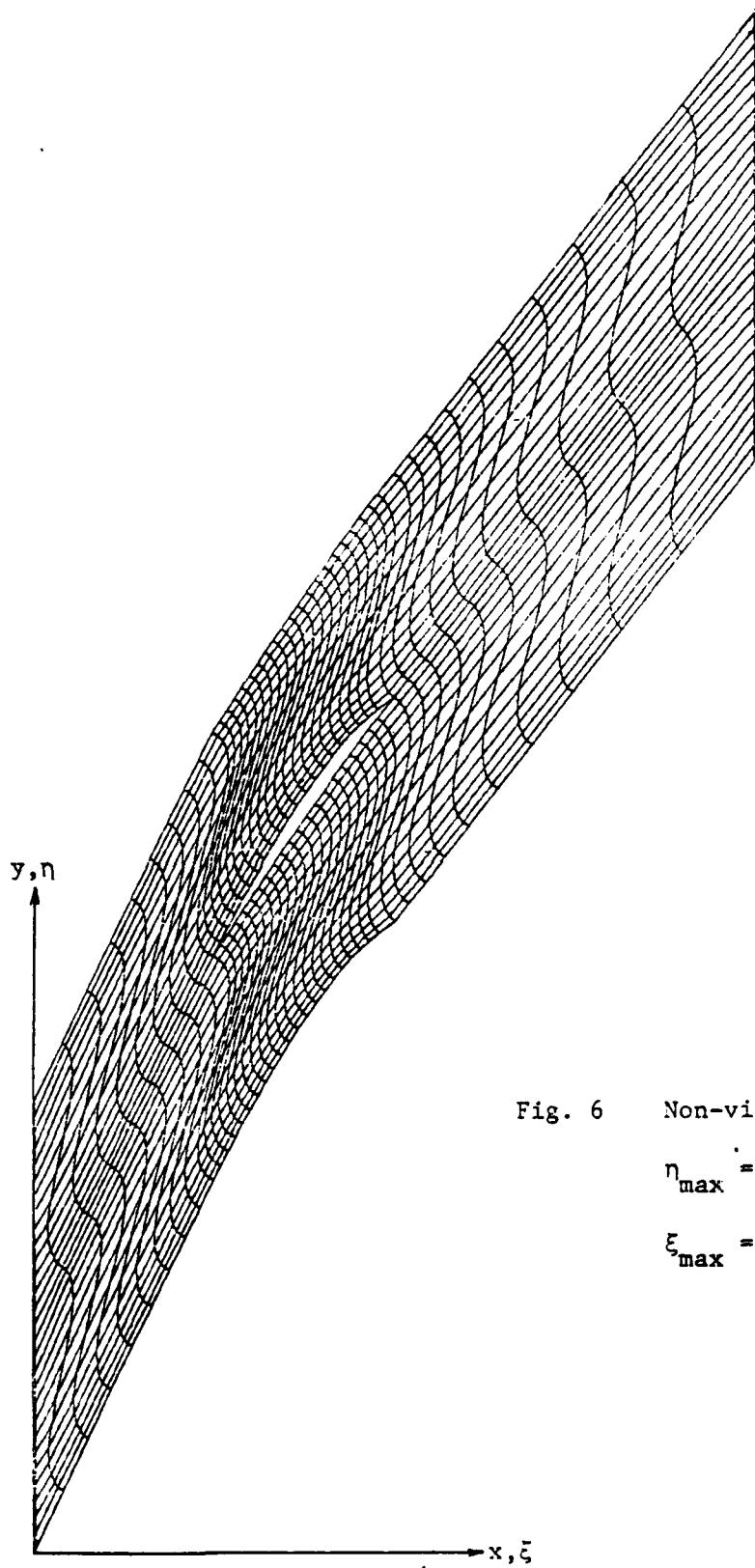


Fig. 6 Non-viscous compressor grid;

$$\eta_{\max} = j_{\max} = 14,$$

$$\xi_{\max} = i_{\max} = 40$$

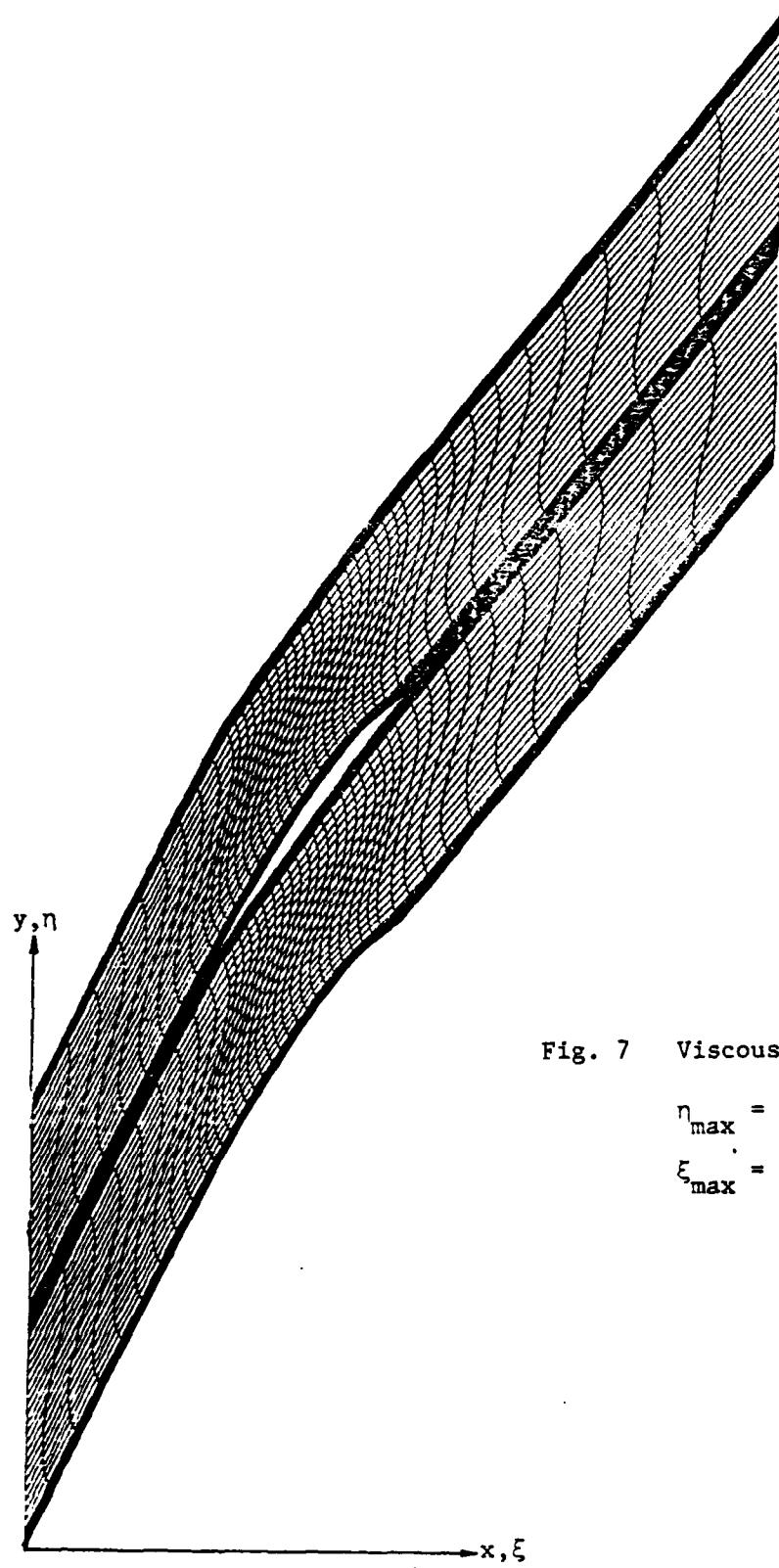


Fig. 7 Viscous compressor grid;

$$\eta_{\max} = j_{\max} = 60,$$

$$\xi_{\max} = i_{\max} = 40$$

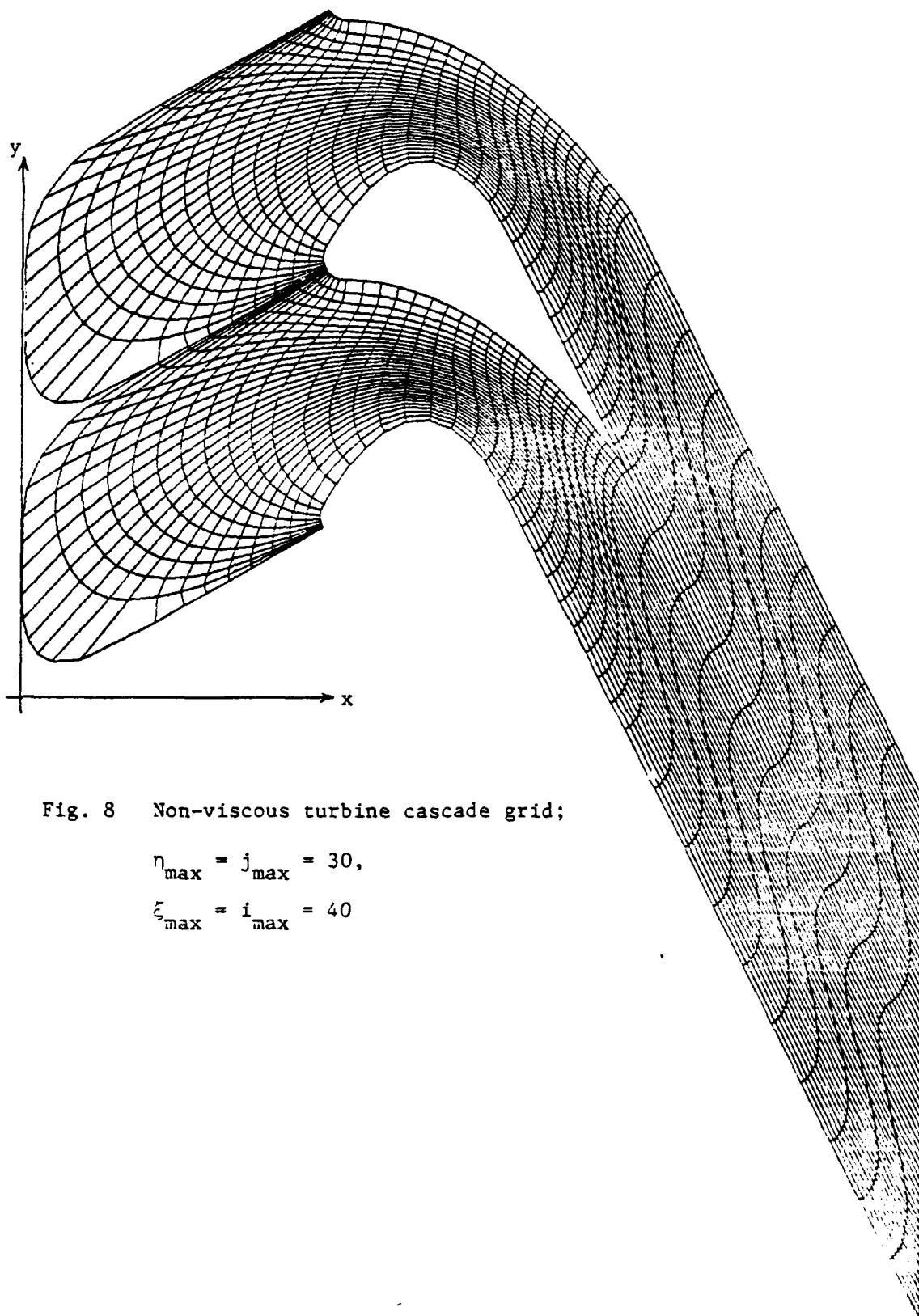
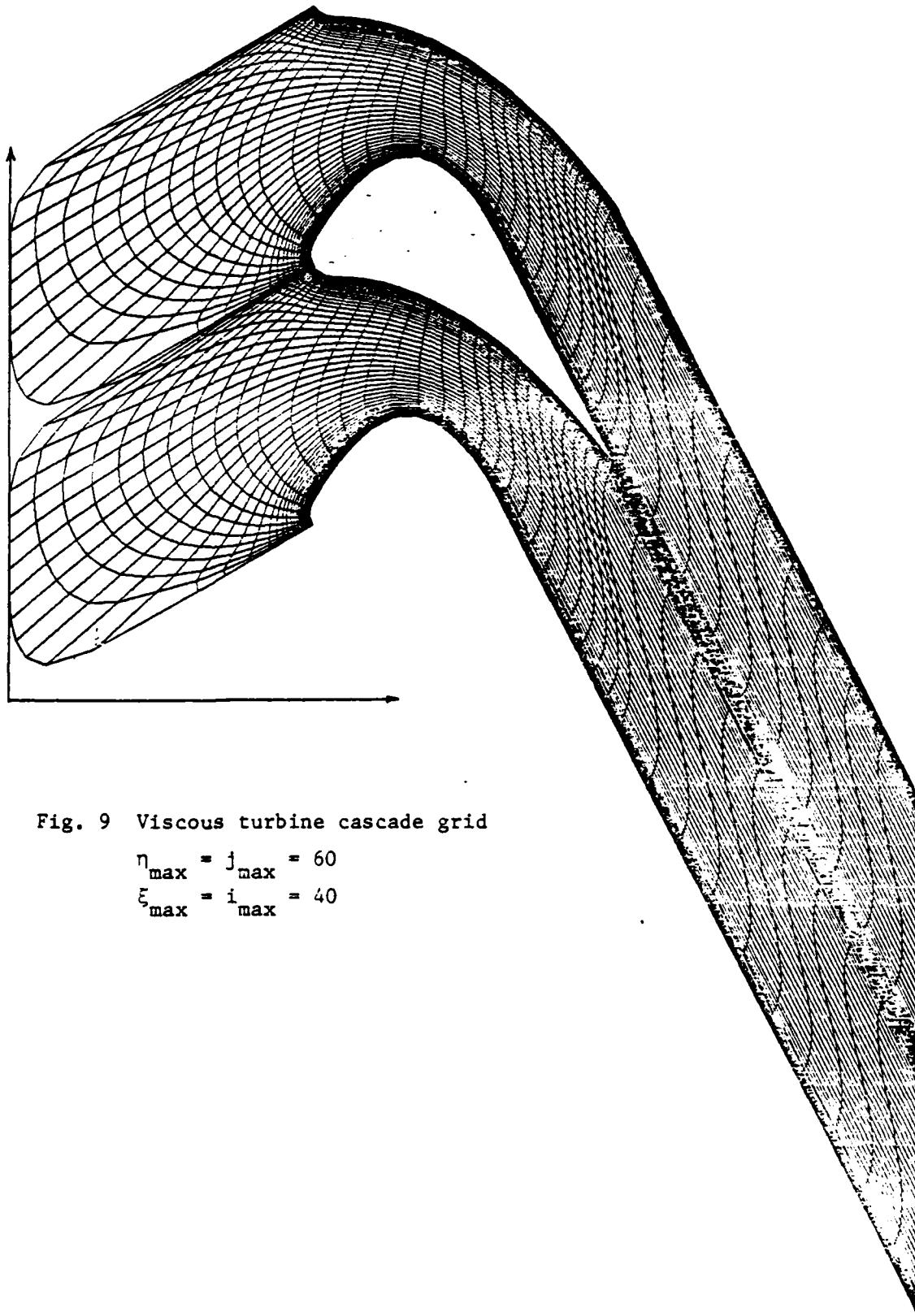


Fig. 8 Non-viscous turbine cascade grid;

$$n_{\max} = j_{\max} = 30,$$

$$\xi_{\max} = i_{\max} = 40$$



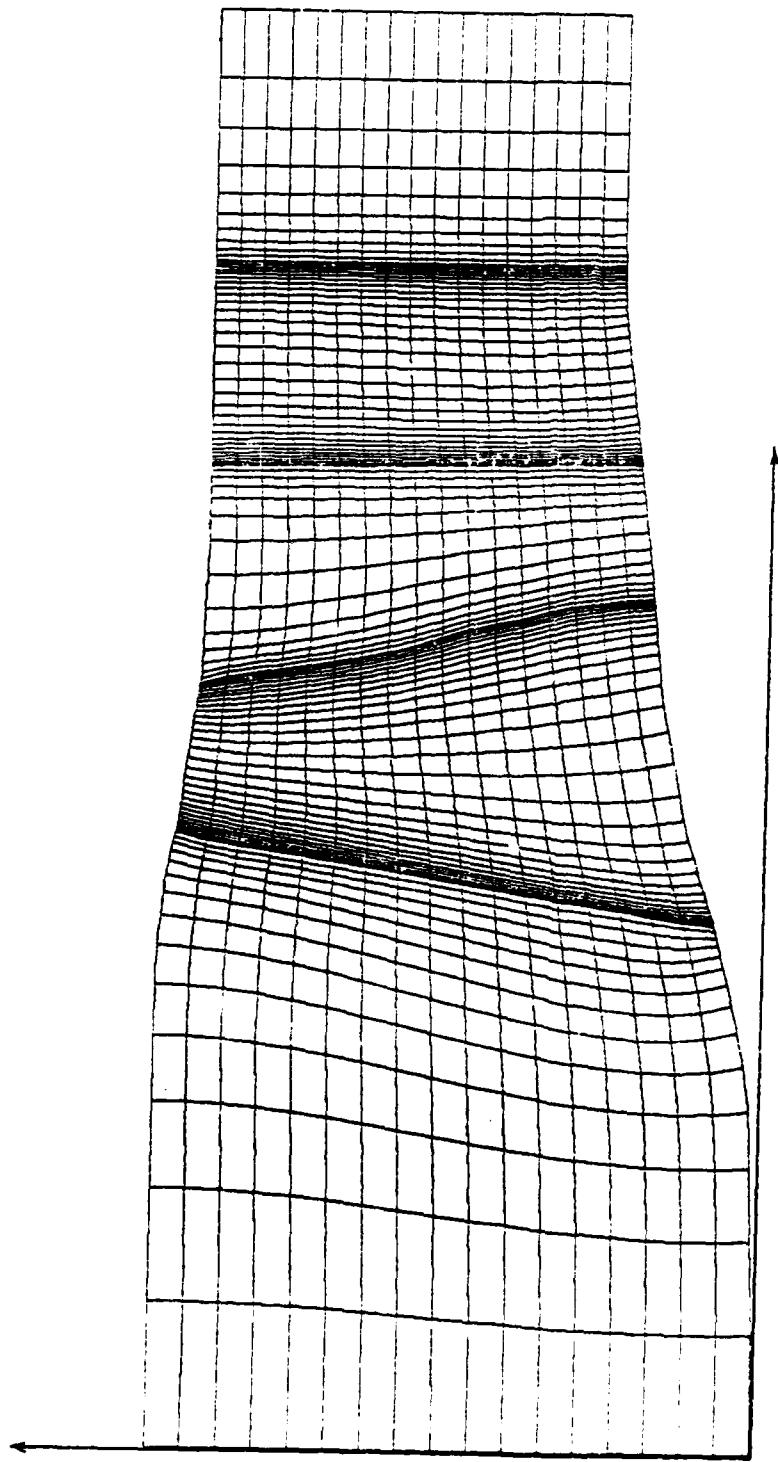


Fig. 10 X-R Coordinates for a rotor-stator calculation

$$\eta_{\max} = j_{\max} = 17 \quad \xi_{\max} = i_{\max} = 98$$

ACE ROTOR, BOGG GRID

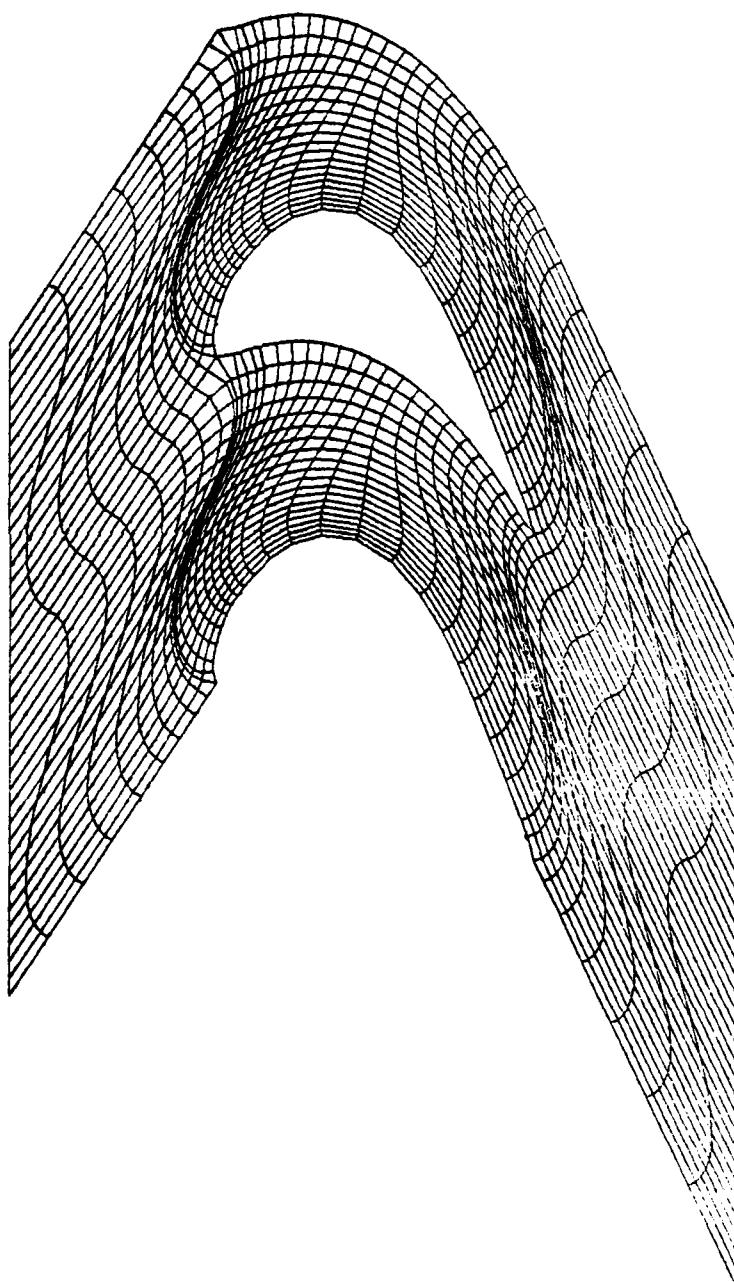


Figure 11

ACE ROTOR BOGG GRID, INVISCID

TOTAL
PRESSURE
ERROR

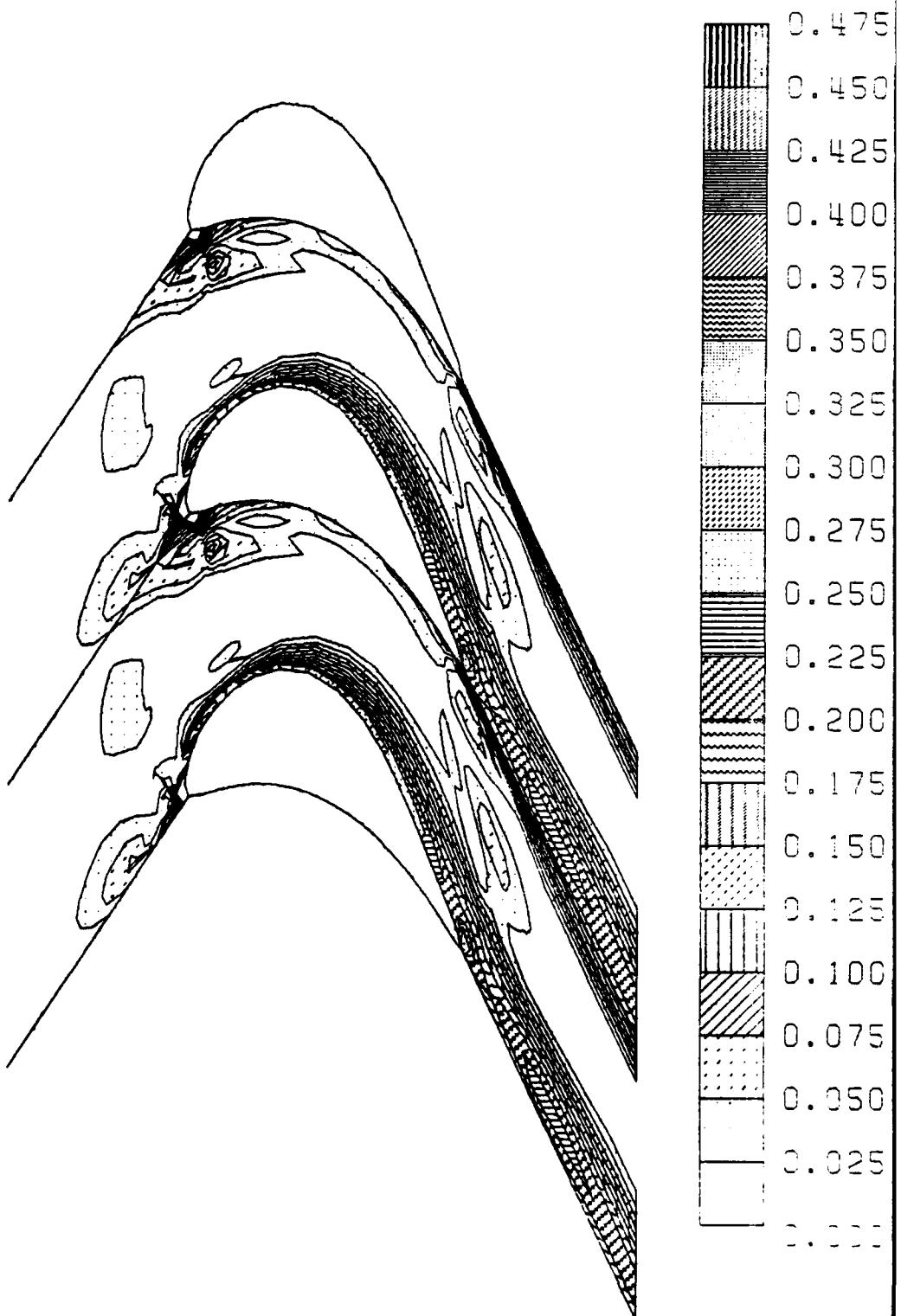
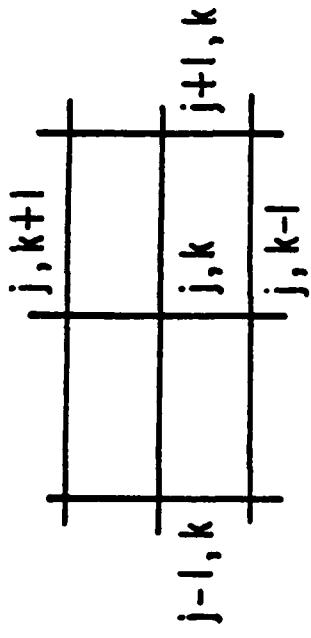


Figure 12

STRONG CONSERVATION LAW FORM

Steady State Solution Operator

79



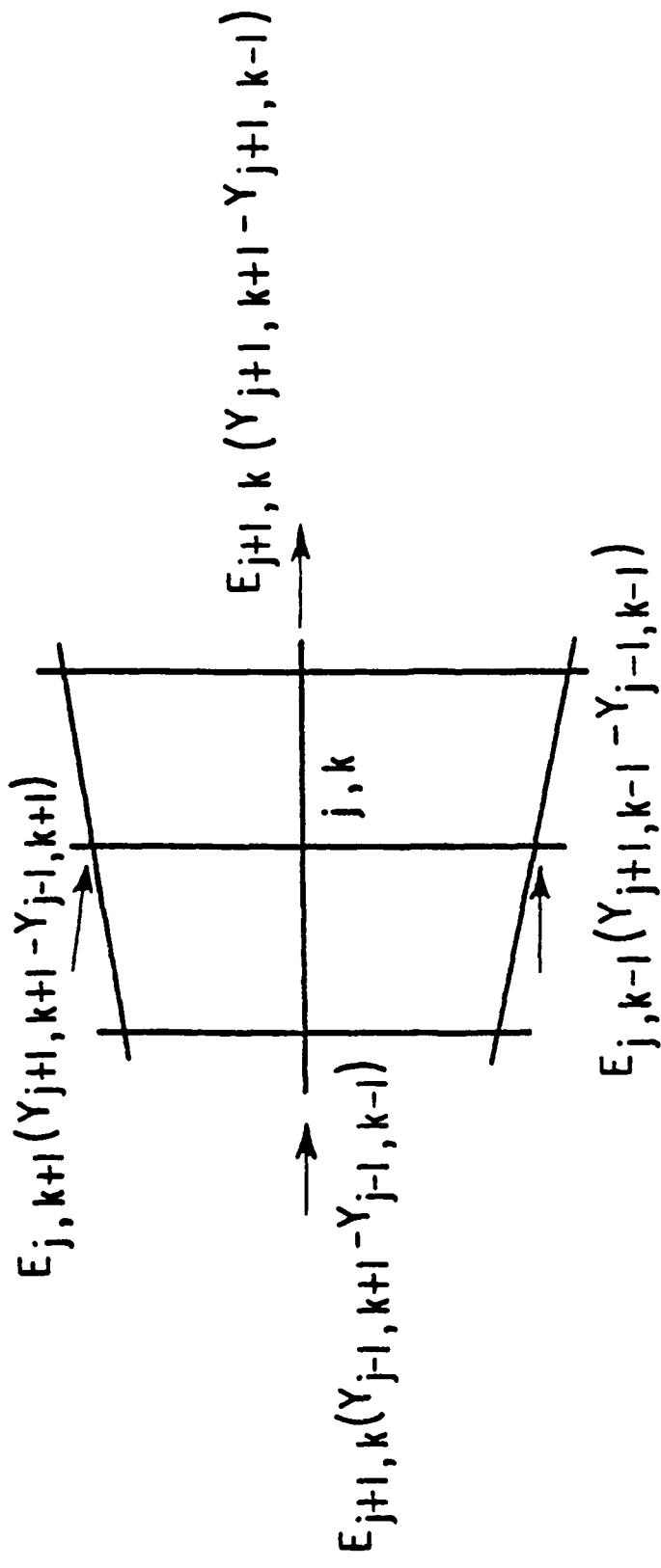
$$\frac{\hat{E}_{j+1,k} - \hat{E}_{j-1,k}}{2\Delta\xi} + \frac{\hat{F}_{j,k+1} - \hat{F}_{j,k-1}}{2\Delta\xi} = 0$$

$$\begin{aligned}\hat{E}_{j+1,k} &= \left[\frac{\xi_X E}{J} \right]_{j+1,k} + \left[\frac{\xi_X F}{J} \right]_{j+1,k}; \hat{F}_{j,k+1} = \left[\frac{\eta_X E}{J} \right]_{j,k+1} + \left[\frac{\eta_Y F}{J} \right]_{j,k+1} \\ \left[\frac{\xi_X}{J} \right]_{j+1,k} &= \frac{Y_{j+1,k+1} - Y_{j+1,k-1}}{\Delta\eta} \quad \left[\frac{\xi_Y}{J} \right]_{j+1,k} = \frac{-[X_{j+1,k+1} - X_{j+1,k-1}]}{\Delta\eta} \\ \left[\frac{\eta_X}{J} \right]_{j,k+1} &= \frac{-[Y_{j+1,k+1} - Y_{j-1,k+1}]}{\Delta\xi} \quad \left[\frac{\eta_Y}{J} \right]_{j,k+1} = \frac{[X_{j+1,k+1} - X_{j-1,k+1}]}{\Delta\xi}\end{aligned}$$

Figure 13

E VECTOR $[\rho u, \rho u^2 + P, \rho u V, \rho u h_0]$ FLUX BALANCE

PHYSICAL SPACE



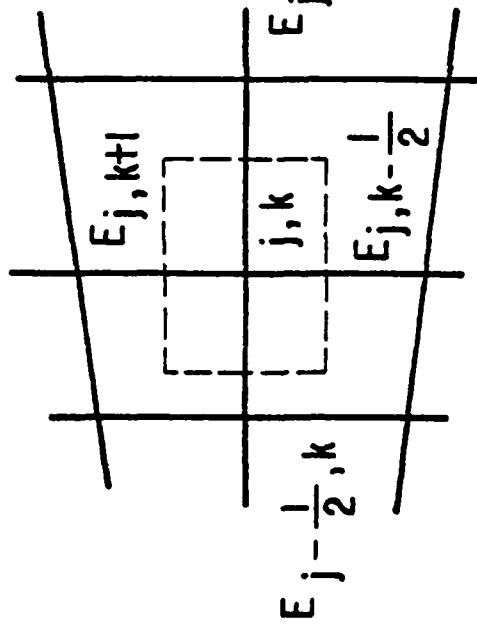
80

- Centered differencing with strong conservation law form and numerical metrics is simple flux balance over 4 cells
- Jacobian defined by central differencing is a poor measure of 4 cell area
- Flux across cell face is not well represented by only point at center of cell face

Figure 14

TRUNCATION ERROR ESTIMATE BY GRID DOUBLING

81



- Situation would be much improved if we could interpret flux balance over 4 cells as flux balance over center cell

ϵ = Difference between 4 cell and 1 cell interpretation

$$\begin{aligned} \epsilon &= (Y_{j,k+1} - Y_{j,k-1}) (E_{j+1,k} - E_{j-1,k}) + (Y_{j+1,k} - Y_{j-1,k}) (E_{j,k+1} - E_{j,k-1}) \\ &\quad + (X_{j+1,k} - X_{j-1,k}) (F_{j,k+1} - F_{j,k-1}) + (X_{j,k+1} - X_{j,k-1}) (F_{j+1,k} - F_{j-1,k}) \end{aligned}$$

Figure 15

ACE ROTOR BOGG GRID INVISCID

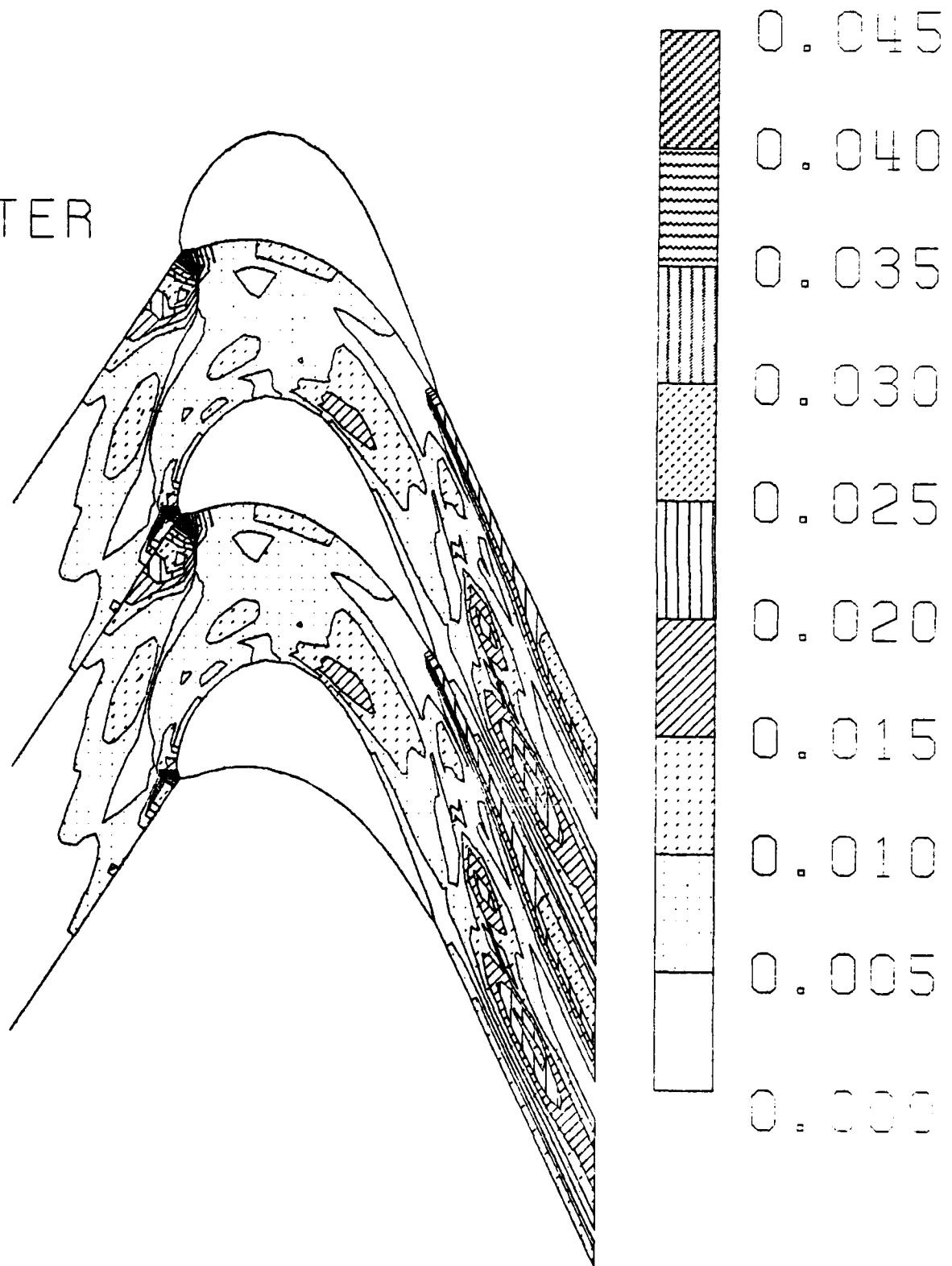
ERROR
PARAMETER

Figure 16

ACE ROTOR SHEARED GRID

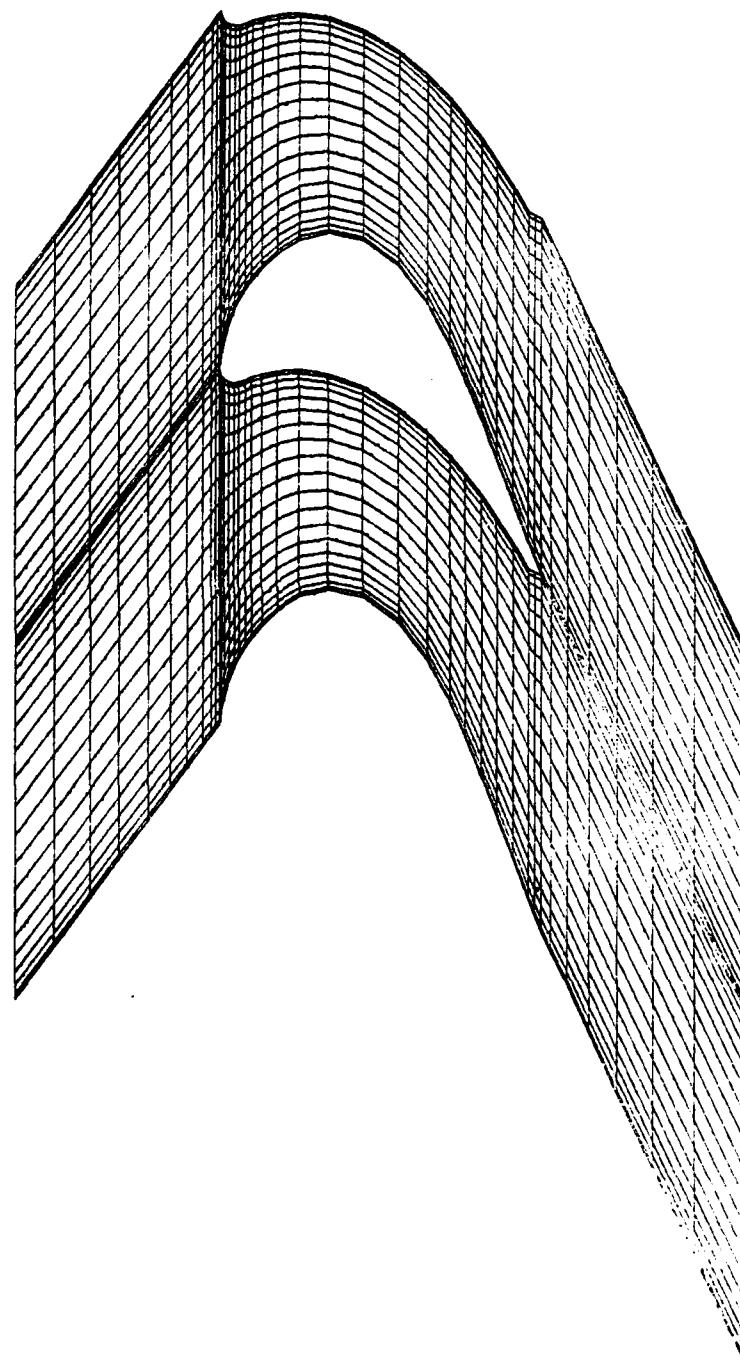


Figure 17

ACE ROTOR, SHEARED GRID, INVISCID

TOTAL
PRESSURE
ERROR

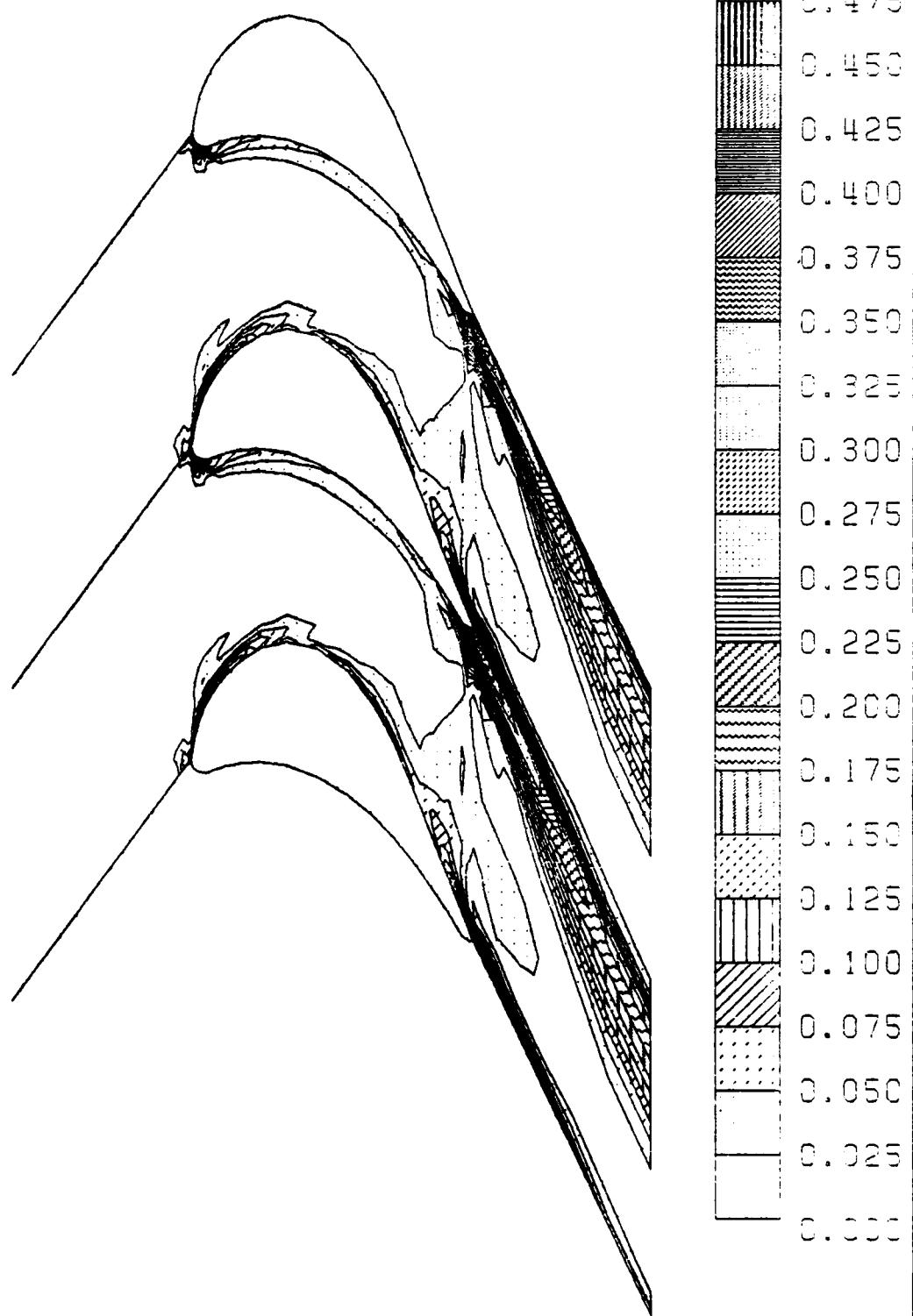


Figure 18

ACE ROTOR, SHEARED GRID, INVISCID

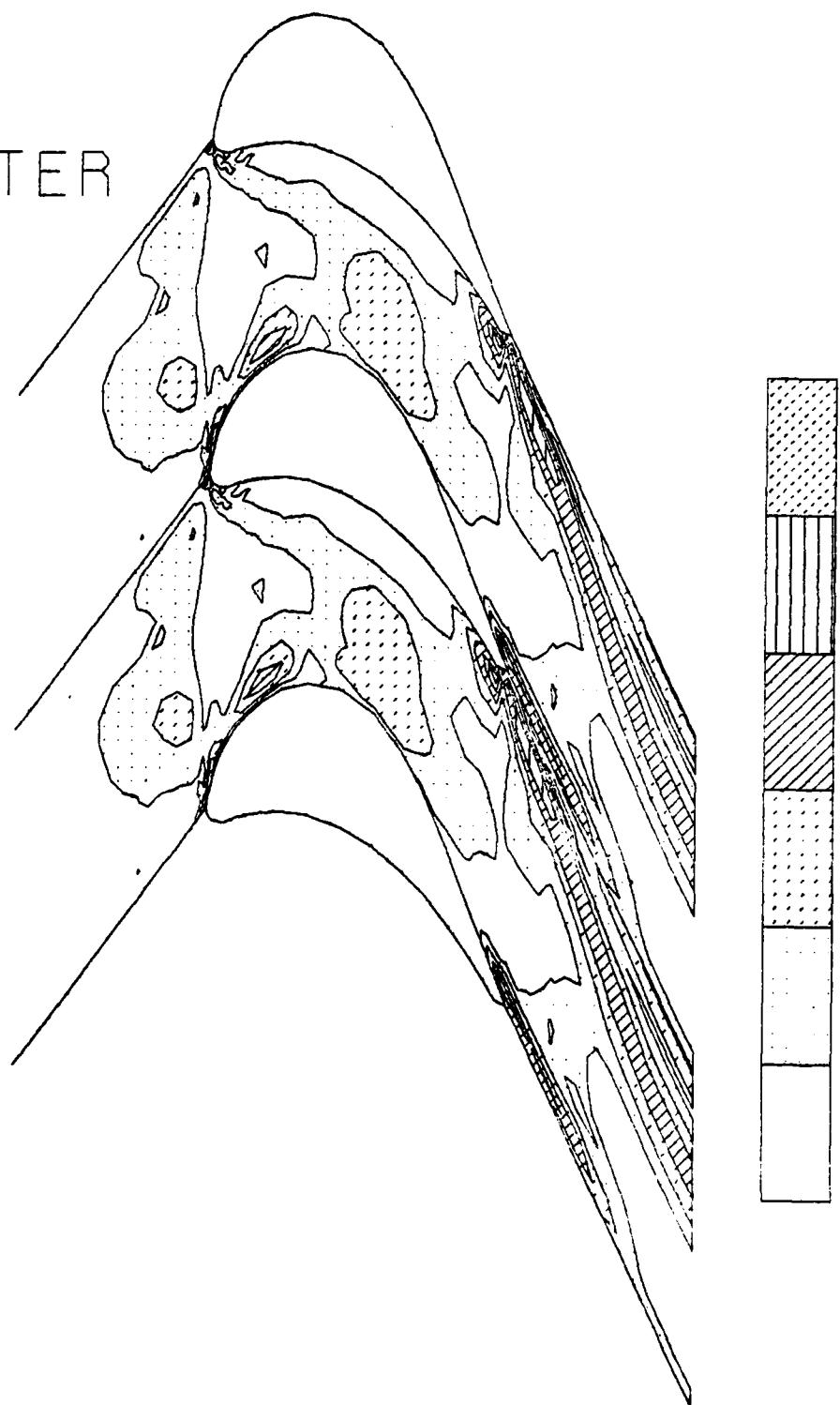
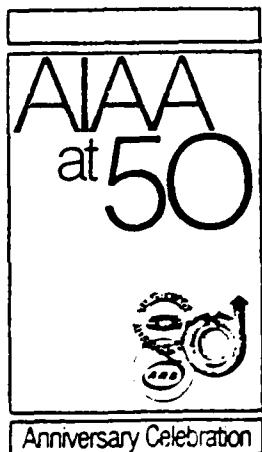
ERROR
PARAMETER

Figure 19

AIAA-82-0063

An Implicit, Bi-Diagonal Numerical Method for
Solving the Navier-Stokes Equations

E. von Lavante and W.T. Thompkins,
Massachusetts Institute of Technology,
Cambridge, MA



AIAA 20th Aerospace
Sciences Meeting

January 11-14, 1982/Orlando, Florida

AN IMPLICIT, BI-DIAGONAL NUMERICAL METHOD
FOR SOLVING THE NAVIER-STOKES EQUATIONS

E. von Lavante*
W.T. Thompkins, Jr.**
Massachusetts Institute of Technology
Cambridge, Massachusetts

Abstract

In recent years, much progress has been made in solving fluid dynamical problems using finite difference methods. Solving inviscid compressible problems in two and three dimensions has become almost routine with many suitable methods, explicit or implicit, available. The problem of compressible, viscous flows in complicated geometries remains, however, a major challenge. Here the fine mesh spacing in the boundary layer region makes the explicit methods with their simple boundary conditions, impractical. Existing implicit methods can make use of large time steps, but require costly inversions of large block-tridiagonal matrices. A method recently developed by MacCormack eliminates this disadvantage by introducing a predictor-corrector scheme requiring the inversion of only block bi-diagonal matrices. It is the aim of present work to extend this method to allow solution of viscous, compressible problems in general coordinates for arbitrary two-dimensional geometries.

Introduction

In recent years, much progress has been made in solving fluid dynamical problems using finite difference methods. Solving inviscid compressible problems in two and three dimensions has become almost routine with many suitable methods, explicit or implicit, available. The problem of compressible, viscous flows in complex geometries remains, however, a major challenge.

The fine mesh spacing required in viscous regions makes the explicit methods with their simple boundary conditions, such as the MacCormack explicit scheme [1,2] impractical. The existing implicit methods, such as Beam and Warming [3] or Pulliam and Steger [4], make the use of large time steps, corresponding to Courant numbers of $0(10^2)$, possible, but require costly inversions of large block-tridiagonal matrices. A method recently developed by MacCormack [5] eliminates this disadvantage by introducing a predictor-corrector scheme requiring the inversion of only block bi-diagonal matrices. The resulting difference equations are either upper or lower block bi-diagonal equations that can be solved easily in one sweep. Unfortunately, this method was demonstrated only for the simple case of flat plate shock-boundary layer interaction. It is the aim of present work to expand this simple implicit method to allow solution of viscous, compressible problems for general two-dimensional geometries.

The present method uses as its first step the explicit predictor-corrector finite difference method of [1], approximating the governing fluid flow equations to second order accuracy in space

and time. Then the second step is used to remove the stability restriction of the first step by transforming the equations of the first step into an implicit form. The resulting matrices are block bi-diagonal and can be easily solved. The Jacobian matrices of the governing flow equations are expressed in a convenient diagonalized form, making any matrix inversion unnecessary. The method was tested on a number of numerical examples, including incompressible and compressible Couette flow and a supersonic diffuser with shock-boundary layer interaction.

Development of Algorithm

The two-dimensional compressible Navier-Stokes equations can be written in the conservation law form

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} + \frac{\partial G(U)}{\partial y} = 0 \quad (1)$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e_t \\ e_t \end{bmatrix} \quad F = \begin{bmatrix} \rho u \\ \rho u^2 + \sigma_x \\ \rho uv + \tau_{xy} \\ (\epsilon_t + \sigma_x)u + \tau_{xy}v - k \frac{\partial T}{\partial x} \end{bmatrix}$$

$$G = \begin{bmatrix} \rho v \\ \rho uv + \tau_{xy} \\ \rho v^2 + \sigma_y \\ (\epsilon_t + \sigma_y)v + \tau_{xy}u - k \frac{\partial T}{\partial y} \end{bmatrix}$$

with

$$p = (\gamma - 1)(e_t - \frac{1}{2}\alpha(u^2 + v^2))$$

$$\sigma_x = p - \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) - 2\mu \frac{\partial u}{\partial x}$$

$$\tau_{xy} = -\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)$$

$$\sigma_y = p - \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) - 2\mu \frac{\partial v}{\partial y}$$

Equation (1) can be expressed in nondimensional form by defining nondimensional variables

$$z' = \frac{z}{z_0} \quad x' = \frac{x}{x_0} \quad y' = \frac{y}{y_0} \quad u' = \frac{u}{a_0} \quad v' = \frac{v}{a_0}$$

$$\tau' = \frac{\tau}{\tau_0} \quad p' = \frac{p}{c_0 a_0^2} \quad e_t' = \frac{e_t}{c_0 a_0^2} \quad u' = \frac{u}{u_0} \quad k' = \frac{k}{k_0}$$

*Now Assistant Professor, Aerospace Eng. Dept.,
Texas A & M, College Station, TX 77843

**Associate Professor, Dept. of Aero and Astro
Member, AIAA

$$t' = t \frac{a_0}{c} \quad Re_0 = \frac{a_0 c l_0}{\mu_0} \quad Pr_0 = \frac{u_0 c p_0}{k_0}$$

The primes will be dropped later on for convenience. The vectors U, F and G then become:

$$U = \begin{pmatrix} p' \\ p'u' \\ p'v' \\ e' \\ t' \end{pmatrix} \quad F = \begin{pmatrix} p'u' \\ p'u'^2 + \sigma'_x \\ p'u'v' + \tau'_{xy} \\ (e'_t + \sigma'_x) u' + \tau'_{xy} v' - k' \frac{1}{Pr_0 Re_0 (\gamma-1)} \frac{\partial T'}{\partial x'} \\ (e'_t + \sigma'_y) v' + \tau'_{xy} u' - k' \frac{1}{Pr_0 Re_0 (\gamma-1)} \frac{\partial T'}{\partial y'} \end{pmatrix}$$

$$G = \begin{pmatrix} p'v' \\ p'u'v' + \tau'_{xy} \\ p'v'^2 + \sigma'_y \\ (e'_t + \sigma'_y) v' + \tau'_{xy} u' - k' \frac{1}{Pr_0 Re_0 (\gamma-1)} \frac{\partial T'}{\partial y'} \end{pmatrix}$$

where

$$\sigma'_x = p' - \frac{\lambda'}{Re_0} \left(\frac{\partial u'}{\partial x'} + \frac{\partial v'}{\partial y'} \right) - 2 \frac{u'}{Re_0} \frac{\partial u'}{\partial x'}$$

$$\tau'_{xy} = - \frac{\mu'}{Re_0} \left(\frac{\partial u'}{\partial y'} + \frac{\partial v'}{\partial x'} \right)$$

$$\sigma'_y = p' - \frac{\lambda'}{Re_0} \left(\frac{\partial u'}{\partial x'} + \frac{\partial v'}{\partial y'} \right) - 2 \frac{u'}{Re_0} \frac{\partial v'}{\partial y'}$$

The equation of state for perfect gas becomes:

$$p' = (\gamma-1) [e'_t - \frac{1}{2} p'(u'^2 + v'^2)]$$

It is also interesting to note that:

$$T' = \frac{Tp'}{p'} \quad ; \quad a' = \sqrt{T'} \quad ; \quad M = M' = \frac{u'}{a'}$$

In most of practical cases it is necessary to transform the Cartesian coordinates x,y into some more general coordinates $\xi(x,y), \eta(x,y)$. The strong conservation law form of (1) can be maintained as shown, for example, by Vinokur [6]. Equation (1) can be then written as

$$\frac{\partial \hat{U}}{\partial t} + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} = 0 \quad (2)$$

where

$$\hat{U} = \frac{U}{J}, \quad \hat{F} = (FE_x + GE_y)/J, \quad \hat{G} = (FE_y + GE_x)/J$$

J is the transformation Jacobian

$$J = \frac{1}{x_\xi y_\eta - x_\eta y_\xi} = \xi_x \eta_y - \xi_y \eta_x \quad (3)$$

The metrics $\xi_x, \xi_y, \eta_x, \eta_y$, etc. are easily formed using the relations

$$\begin{aligned} \xi_x &= Jy_\eta & \xi_y &= -Jx_\eta \\ \eta_x &= -Jy_\xi & \eta_y &= Jx_\xi \end{aligned} \quad (4)$$

The method of numerical integration of equation (2) has been adopted from MacCormack [5], where it is explained in detail. The resulting integration scheme will be therefore presented without detailed development. Equation (2) is integrated by the following implicit predictor-corrector set of finite difference equations:

Predictor:

$$\begin{aligned} \hat{U}_{i,j}^n &= -\Delta t \left[\frac{\Delta^+ \hat{F}_{i,j}^n}{\Delta \xi} + \frac{\Delta^+ \hat{G}_{i,j}^n}{\Delta \eta} \right] \\ \left[I - \Delta t \frac{\Delta^+ |\hat{A}|^n}{\Delta \xi} \right] \left[I - \Delta t \frac{\Delta^+ |\hat{B}|^n}{\Delta \eta} \right] \hat{U}_{i,j}^{n+1} &= \hat{U}_{i,j}^n \\ \hat{U}_{i,j}^{n+1} &= \hat{U}_{i,j}^n + \delta \hat{U}_{i,j}^{n+1} \end{aligned}$$

Corrector:

$$\begin{aligned} \hat{U}_{i,j}^{n+1} &= -\Delta t \left[\frac{\Delta^- \hat{F}_{i,j}^{n+1}}{\Delta \xi} + \frac{\Delta^- \hat{G}_{i,j}^{n+1}}{\Delta \eta} \right] \\ \left[I + \Delta t \frac{\Delta^- |\hat{A}|^{n+1}}{\Delta \xi} \right] \left[I + \Delta t \frac{\Delta^- |\hat{B}|^{n+1}}{\Delta \eta} \right] \hat{U}_{i,j}^{n+1} &= \hat{U}_{i,j}^{n+1} \\ \hat{U}_{i,j}^{n+1} &= \frac{1}{2} \left[\hat{U}_{i,j}^n + \hat{U}_{i,j}^{n+1} + \delta \hat{U}_{i,j}^{n+1} \right] \end{aligned} \quad (5)$$

where $|\hat{A}|, |\hat{B}|$ are matrices with positive eigenvalues, related to the Jacobians $A = (\partial \hat{U} / \partial U)$ and $B = (\partial \hat{U} / \partial \eta)$, $(\Delta^+ / \Delta \xi)$ and $(\Delta^+ / \Delta \eta)$ are one-sided forward differences and $(\Delta^- / \Delta \xi)$ and $(\Delta^- / \Delta \eta)$ are one-sided backward differences.

The Jacobians \hat{A} and \hat{B} are related to the Jacobians $A = \partial F / \partial U$ and $B = \partial G / \partial U$ by

$$\begin{aligned} \hat{A} &= A \xi_x + B \xi_y \\ \hat{B} &= A \eta_x + B \eta_y \end{aligned} \quad (6)$$

and are given in, for example, Steger [7].

$\hat{A} =$

$$\begin{vmatrix} 0 & \xi_x & \xi_y & 0 \\ -u \xi_\xi + \xi_x \beta a & U_\xi - (\beta-1) \xi_u & \xi_y u - B \xi_x v & B \xi_x \\ v \xi_\xi + \xi_y \beta a & -B \xi_u + \xi_x v & U_\xi - (\beta-1) \xi_v & B \xi_y \\ A_{41} & A_{42} & A_{43} & (B+1) U_\xi \end{vmatrix} \quad (7)$$

where

$$A_{41} = U_\xi [\alpha(\beta-1) - \frac{a^2}{B}], \quad A_{42} = -B U_\xi u + \left(\frac{a^2}{B} + \alpha \right) \xi_x;$$

$$\text{and } A_{43} = -B U_\xi v + \left(\frac{a^2}{B} + \alpha \right) \xi_y.$$

here $\alpha = \frac{1}{2}(u^2 + v^2)$, $\beta = (\gamma - 1)$, $a = \sqrt{\gamma(p/\rho)}$, $\xi_x = \partial E/\partial x$, $\xi_y = (\partial E/\partial y)$ and U_ξ , U_η are the contravariant velocities

$$U_\xi = u\xi_x + v\xi_y$$

$$U_\eta = u\eta_x + v\eta_y$$

where $\eta_x = 3n/3x$ and $\eta_y = 3n/3y$. The Jacobian matrix \hat{B} is obtained by substituting U_ξ , ξ_x , ξ_y by U_η , η_x , η_y .

The integration scheme (5) can be much simplified by diagonalizing \hat{A} and \hat{B} . Once the eigenvalues of \hat{A} and \hat{B} are known, it is possible to express A and B in the form

$$\hat{A} = \hat{S}_\xi \Lambda_A \hat{S}_\xi^{-1}$$

$$\hat{B} = \hat{S}_\eta \Lambda_B \hat{S}_\eta^{-1} \quad (8)$$

where Λ_A and Λ_B are diagonal matrices consisting of the eigenvalues of \hat{A} , $\lambda_{A,1}, \dots, \lambda_{A,4}$, and \hat{B} , $\lambda_{B,1}, \dots, \lambda_{B,4}$, respectively. The vectors \hat{S}_ξ and \hat{S}_η are constructed using the eigenvectors of \hat{A} and \hat{B} , respectively, as columns.

Warming et al [8] found the eigenvalues and eigenvectors of the nonconservative system corresponding to (1)

$$\frac{\partial \tilde{U}}{\partial t} + \tilde{A} \frac{\partial \tilde{U}}{\partial x} + \tilde{B} \frac{\partial \tilde{U}}{\partial y} = 0$$

They give the eigenvector matrix T such that if $\tilde{P} = k_1 \tilde{A} + k_2 \tilde{B}$, then

$$T^{-1} \tilde{P} T = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \\ 0 & \lambda_3 \\ 0 & \lambda_4 \end{pmatrix} = \Lambda$$

for arbitrary k_1, k_2 . The matrix T is:

$$T = \begin{pmatrix} 1 & 0 & \frac{0}{\sqrt{2}a} & \frac{0}{\sqrt{2}a} \\ 0 & K_2 & \frac{K_1}{\sqrt{2}} & \frac{-K_1}{\sqrt{2}} \\ 0 & -K_1 & \frac{K_2}{\sqrt{2}} & \frac{-K_2}{\sqrt{2}} \\ 0 & 0 & \frac{c_1}{\sqrt{2}} & \frac{0a}{\sqrt{2}} \end{pmatrix} \quad \begin{aligned} K_1 &= k_1 / \sqrt{k_1^2 + k_2^2} \\ K_2 &= k_2 / \sqrt{k_1^2 + k_2^2} \end{aligned}$$

The Jacobians A and B of the conservative system (1) are obtained from the nonconservative \tilde{A} and \tilde{B} by

$$\tilde{A} = M^{-1} A M \quad \tilde{B} = M^{-1} B M \quad (9)$$

where

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ u & 0 & 0 & 0 \\ v & 0 & 0 & 0 \\ a & au & bv & \frac{1}{3} \end{pmatrix}$$

If now $P = k_1 \tilde{A} + k_2 \tilde{B}$ for the conservative system then $\tilde{P} = M^{-1} P M$ gives the proper relation for conversion of P from conservative to nonconservative. Both P and \tilde{P} have the same eigenvalues, so that

$$P = M T \Lambda (M T)^{-1} \quad (10)$$

Equation (10) can be used to find the matrices \hat{S}_ξ , Λ_A , \hat{S}_ξ^{-1} and S_η , Λ_B , \hat{S}_η^{-1} in (8). One can, for example, set the arbitrary k_1 and k_2 to $k_1 = \xi_x$, $k_2 = \xi_y$ and obtain

$$P = \xi_x A + \xi_y B = \hat{A}$$

The comparison of (10) and (8) leads to

$$\hat{S}_\xi = M T \quad (11)$$

\hat{S}_ξ can be found by carrying out the multiplication in (11). The result is

$$\hat{S}_\xi =$$

$$\begin{pmatrix} 1 & 0 & : & \frac{0}{\sqrt{2}} & : & \frac{0}{\sqrt{2}} \\ : & : & : & : & : & : \\ u & \frac{\partial \xi_y}{c_1} & : & \rho \left(\frac{u}{a} + \frac{\xi_x}{c_1} \right) / \sqrt{2} & : & \rho \left(\frac{u}{a} - \frac{\xi_x}{c_1} \right) / \sqrt{2} \\ : & : & : & : & : & : \\ v & \frac{-\partial \xi_x}{c_1} & : & \rho \left(\frac{v}{a} + \frac{\xi_y}{c_1} \right) / \sqrt{2} & : & \rho \left(\frac{v}{a} - \frac{\xi_y}{c_1} \right) / \sqrt{2} \\ : & : & : & : & : & : \\ a & \frac{\rho(u\xi_y - v\xi_x)}{c_1} & : & \rho \left(\frac{a}{a} + \frac{\xi_x}{c_1} + \frac{a}{3} \right) / \sqrt{2} & : & \rho \left(\frac{a}{a} - \frac{\xi_x}{c_1} + \frac{a}{3} \right) / \sqrt{2} \end{pmatrix} \quad (12)$$

where $c_1 = \sqrt{\xi_x^2 + \xi_y^2}$. \hat{S}_ξ^{-1} is found simply from $\hat{S}_\xi^{-1} = T^{-1} M^{-1}$

$$\hat{S}_\xi^{-1} = \begin{pmatrix} 1 - \frac{\Delta t}{a^2} & \frac{\Delta u}{a^2} & \frac{\Delta v}{a^2} & -\frac{3}{a^2} \\ -u\xi_v + v\xi_u & \frac{\xi_v}{a c_1} & -\frac{\xi_x}{a c_1} & 0 \\ \frac{u^2}{a c_1} & \frac{\xi_x}{c_1} - \frac{u}{a} & \frac{\xi_y}{c_1} - \frac{v}{a} & \frac{3}{a^2} \\ \frac{u^2}{a^2 c_1} & \frac{1}{a^2} & \frac{1}{a^2} & \frac{3}{a^2} \\ \frac{\alpha \beta}{a^2} + \frac{u \xi}{a c_1} & -\left(\frac{\xi_x}{c_1} - \frac{u}{a}\right) & -\left(\frac{\xi_y}{c_1} + \frac{v}{a}\right) & \frac{3}{a^2} \\ \frac{\alpha \beta}{a^2} & \frac{1}{a^2} & \frac{1}{a^2} & \frac{3}{a^2} \end{pmatrix} \quad (13)$$

And finally $\hat{\Lambda}_A$ is formed from eigenvalues of \hat{A} . These are given by, for example, Steger [7]:

$$\hat{\Lambda}_A = \begin{pmatrix} U_\xi & 0 & 0 & 0 \\ 0 & U_\xi & 0 & 0 \\ 0 & 0 & U_\xi + ac_1 & 0 \\ 0 & 0 & 0 & U_\xi - ac_1 \end{pmatrix} \quad (14)$$

\hat{S}_n , $\hat{\Lambda}_B$ and \hat{S}_n^{-1} are similar; they can be found by replacing ξ_x , ξ_y and U_ξ by η_x , η_y and U_n . Following closely the MacCormack approach in [5], the matrices $|\hat{A}|$ and $|\hat{B}|$ in (5) are formed by replacing the matrices Λ_A and Λ_B by positively valued diagonal matrices D_A and D_B such that

$$\begin{aligned} D_A &= |\Lambda_A| + \lambda_A I \\ D_B &= |\Lambda_B| + \lambda_B I \end{aligned} \quad (15)$$

where

$$\lambda_A = \frac{2v}{\rho \Delta \xi} \left(\xi_x^2 + \xi_y^2 \right) - \frac{1}{2} \frac{\Delta \xi}{\Delta t}$$

$$\lambda_B = \frac{2v}{\rho \Delta n} \left(\eta_x^2 + \eta_y^2 \right) - \frac{1}{2} \frac{\Delta n}{\Delta t}$$

and

$$v = \max \left(\frac{u}{Re_0}, \frac{(\lambda + 2u)}{Re_0}, \frac{ky}{Pr_0 Re_0} \right)$$

Formula (15) for D_A and D_B assumes that viscous effects are modeled in the implicit part of the scheme by addition of the terms λ_A , λ_B which include viscosity through the coefficient v . The elements of D_A , D_B are non-negative; negative values of the elements of D_A , D_B mean that the CFL condition is met and there is no need to use the implicit portion of the scheme (5). The integration scheme (5) can now be carried out in the following steps:

1) let $\hat{U}_{i,j}^n = -\Delta t \left(\frac{\Delta F_{i,j}^n}{\Delta \xi} + \frac{\Delta G_{i,j}^n}{\Delta n} \right)$ at every point inside the calculational domain.

2) solve

$$\left(I - \Delta t \frac{\Delta^+ |\hat{A}|^n}{\Delta \xi} \right) \left(I - \Delta t \frac{\Delta^+ |\hat{B}|}{\Delta n} \right) \hat{U}_{i,j}^{n+1} = \hat{U}_{i,j}^n \text{ for } \hat{U}_{i,j}^{n+1}.$$

This is done in two steps by denoting $\hat{U}_{i,j}^* = \left(I - \Delta t \frac{\Delta^+ |\hat{B}|}{\Delta n} \right) \hat{U}_{i,j}^{n+1}$, resulting in two vector equations.

$$a) \left(I - \Delta t \frac{\Delta^+ |\hat{A}|^n}{\Delta \xi} \right) \hat{U}_{i,j}^* = \hat{U}_{i,j}^n$$

$$b) \left(I - \Delta t \frac{\Delta^+ |\hat{B}|}{\Delta n} \right) \hat{U}_{i,j}^{n+1} = \hat{U}_{i,j}^*$$

A closer look at step 2a reveals that it is an upper bi-diagonal equation that can be solved by sweeping in decreasing ξ direction for a constant n . Substituting (8) in step 2a and some simple manipulation leads to

$$\begin{aligned} &\left(I + \Delta t \hat{S}_\xi \hat{D}_A \hat{S}_\xi^{-1} \right) \hat{U}_{i,j}^* \\ &= \hat{U}_{i,j}^n + \Delta t |\hat{A}|_{i+1,j} \hat{U}_{i+1,j}^* \end{aligned} \quad (16)$$

Denoting $w = \hat{U}_{i,j}^n + \Delta t |\hat{A}|_{i+1,j} \hat{U}_{i+1,j}^*$ and some matrix multiplication gives finally

$$\hat{U}_{i,j}^* = \hat{S}_\xi (I + \Delta t D_A)^{-1} \hat{S}_\xi^{-1} w \quad (17)$$

Equation (17) can be very easily solved since \hat{S}_ξ and \hat{S}_ξ^{-1} are known and the inversion of the diagonal ξ matrix $(I + \Delta t D_A)$ is trivial. After all $\hat{U}_{i,j}^*$ inside the calculational domain are determined the step 2b can be carried out in the same manner. The corrector steps are analogous to the predictor steps. The major problem in the above described scheme is finding the proper boundary values of the expression $\Delta t |\hat{A}|_{i,j} \hat{U}_{i,j}^*$ for $i = i_{\max}$, $i = 1$ and $j = j_{\max}$ and $j = 1$, because these are not known at the time when the sweep begins. They will be dealt with in the next section.

Boundary Conditions

Since the present method uses upwind spatial derivatives, a starting value of the expression $\Delta t |\hat{A}| \hat{U}$ or $\Delta t |\hat{B}| \hat{U}$ is needed for both the predictor and corrector. For simplicity, we will devote these expressions \hat{W} ; they represent the implicit part of the boundary conditions. The explicit boundary conditions, needed for evaluation of the expression ΔU^n , are obtainable more easily. At the boundary, equation (16) will take the following form:

$$(I + \Delta t \hat{S}_\xi \hat{D}_A \hat{S}_\xi^{-1}) \hat{U}_{i,j}^* = \hat{U}_{i,j}^n + \hat{W} \quad (18)$$

In order to maintain the unconditional stability of scheme (5), the value of δW would have to be evaluated implicitly. An improper treatment of the implicit part of the boundary conditions, such as lagging in time, will limit the stability region of (5) to CFL members of 0 (1). The present work developed some ad hoc procedures for determining the value of δW that gave satisfactory results for the test cases used. A more vigorous treatment of the boundary conditions for the present method will be required to develop their generally valid formulation.

Following boundary conditions were implemented:

a) Supersonic inflow boundary: At these boundaries, all the eigenvalues of A are positive, so that all characteristics point from outside into the computational domain. All the elements of U are therefore specified at this boundary. For time-independent boundary conditions one obtains: $p, p_u, p_v, e = \text{constant}; \delta W = 0$.

b) Supersonic outflow boundary: Here all the characteristics have the direction form inside to outside. All the values of U must be therefore extrapolated from the computational domain. The explicit part of the boundary conditions does not represent any problems; the elements of U are in this case linearly extrapolated from the computational domain:

$$\hat{U}_{i_{\max}} = 2\hat{U}_{i_{\max}-1} - \hat{U}_{i_{\max}-2} \quad (19)$$

The same extrapolation was applied to δW .

$$\delta W = 2\Delta t (|\hat{A}| \delta \hat{U})_{i_{\max}-1} - \hat{U}_{i_{\max}-2} \quad (20)$$

Since, at the time of evaluation of δW , the expressions on the left-hand side of (20) are not known, δW would have to be calculated by using an implicit scheme at the three points $i_{\max}, i_{\max}-1, i_{\max}-2$.

In the present work the CFL number in the x -direction was always less than 1, so that purely explicit boundary conditions were used, giving (19) and $\delta W = 0$.

c) Solid wall boundary: The wall was placed between the first and second grid point and reflective boundary conditions were used. The explicit boundary conditions for an adiabatic wall are then:

$$\begin{aligned} \rho u_{i,1} &= -\rho u_{i,2} & \rho v_{i,1} &= -\rho v_{i,2} \\ \rho_{i,1} &= -\rho_{i,2} & e_{i,1} &= e_{i,2} \end{aligned}$$

The corrector value of δW can be obtained using the same principle from the predictor value of $\Delta t |\hat{B}| \delta \hat{U}$ at the point $j = 2$

$$\delta W^{n+1} = R \cdot \Delta t (|\hat{B}| \delta \hat{U})_{j=2}^{n+1} \quad (21)$$

where

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

There still remains the problem of finding the predictor value of W . The present work used the following procedure:

$$\delta W^{n+1} = R \cdot \Delta t (|\hat{B}| \delta \hat{U})_{j=2}^{n+1} = \delta W^n$$

while switching the direction of the predictor sweep after every completed predictor-corrector sequence. This is equivalent to lagging the boundary value of change of flux one half time step.

d) Periodic boundaries: The periodic boundaries were again placed between the first and second grid point, and δW was lagged one half step. Placing the periodic boundaries at, for example, $i = 1 + (1/2)$ and $i = i_{\max} - (1/2)$ resulted in the following scheme: For predictor sweeping in direction of increasing i ,

$$\overline{\rho}_{1,j}^{n+1} = \overline{\rho}_{i_{\max}-1,j}^n : \overline{u}_{1,j}^{n+1} = \overline{u}_{i_{\max}-1,j}^n$$

$$\overline{v}_{1,j}^{n+1} = \overline{v}_{i_{\max}-1,j}^n : \overline{e}_{1,j}^{n+1} = \overline{e}_{i_{\max}-1,j}^n \text{ and}$$

$$\overline{\delta W}_{1,j}^{n+1} = \overline{\delta W}_{i_{\max}-1,j}^n$$

and for corrector,

$$\overline{\rho}_{i_{\max},j}^{n+1} = \overline{\rho}_{2,j}^{n+1}, \overline{u}_{i_{\max},j}^{n+1} = \overline{u}_{2,j}^{n+1},$$

$$\overline{v}_{i_{\max},j}^{n+1} = \overline{v}_{2,j}^{n+1}, \overline{e}_{i_{\max},j}^{n+1} = \overline{e}_{2,j}^{n+1} \text{ and}$$

$$\overline{\delta W}_{i_{\max},j}^{n+1} = \overline{\delta W}_{2,j}^{n+1}.$$

Here, the grid lines are assumed to be orthogonal at these boundaries.

No attempt was made at this time to create nonreflective implicit boundary conditions based on the characteristics of (5). Note also that the outflow boundary conditions are only explicit. The present method is therefore limited to supersonic inflow and supersonic outflow boundaries with CFL number 0(1) in the outflow direction or periodic inflow/outflow.

Examples

The present numerical method was tested on two examples, the Couette flow and a supersonic diffuser.

A) Couette flow: The physical domain consisted of stationary lower wall at $j = 1$, uniformly moving upper wall at $j = 16$ and two periodic boundaries at $i = 1 + (1/2)$ and $i = 10 + (1/2)$. The computational domain had 16 grid lines in the η -direction and 11 grid lines in the ξ -direction. Several cases were run for Reynolds numbers based on the distance between the upper and lower wall between 6.2 and

2.3×10^3 at Mach numbers between 0.09 and 0.75. Courant numbers in η -direction of up to 1000 did not cause any problems, but the Courant number in ξ -direction was limited to 0(1) by the periodic boundary conditions. Depending on the Reynolds number and the Courant number, the steady state solution was reached after 10 to 500 iterations. The results for $U_{up} = 0.75$ are shown in Fig. 1.

B) Supersonic diffuser: The supersonic diffuser flow calculation was performed to test the present method on a more demanding case for which analytical and experimental data are available. The diffuser had the shape shown in Fig. 2. It has a straight lower wall and upper wall with a compression corner to produce a shock of required strength. The orthogonal grid shown in Fig. 2 was numerically generated. There were 51 grid lines in η -direction and 51 grid lines in ξ -direction, with 20 grid lines in the viscous layer region close to the wall. Upstream, the Mach number was 2, and the velocity distribution corresponded to $Re_x = 1.25 \times 10^4$. The pressure ratio across the shock $P_{x_1}/P_{x_0} = 1.2$; it was chosen such that direct comparison with experimental results [9] was possible. The boundary layer was assumed to be laminar. At the reflection point the Reynolds number was $Re_x = 3 \times 10^5$; the pressure increase due to the shock caused boundary layer separation. The resulting pressure profiles after 600 iterations at CFL number $C_f = 160$ agree rather well with experimental data from Hakkinen et al [9] (see Fig. 4). The pressures at the walls display slightly higher values than the experiment due to the blockage effect of the boundary layer. The c_f coefficient in Fig. 4 shows again good agreement with analytical data by Bush [10] and experimental data which were performed on a flat plate.

Some problems were experienced with the boundary conditions in η -direction. Best results were obtained by switching sweep directions after one complete time step. The corrector value of δW was reflected off the opposite wall. Although the method has natural dissipation, the steep gradient at the shock boundary layer interaction region caused sometimes stability problems. A weak fourth order explicit damping term was added to the right-handed side term, eliminating the instability.

Conclusion

The present method offers substantial potential for use in complex compressible viscous flow calculations. Using the same Courant numbers, it is faster and simpler than existing implicit methods because it does not require inversion of block tridiagonal matrices. At the present time its use at higher Courant numbers is limited by the choice of boundary conditions to supersonic flows. Its general usefulness for transonic flows depends on future research in the area of boundary conditions.

The implicit part of the boundary conditions deserves special attention as well as the formulation of non-reflective boundary conditions for the inflow and outflow boundaries.

The $\xi = \text{constant}$ boundaries (upstream and downstream) did not represent a problem in this case, because the CFL number was here 0(1).

References

- [1] MacCormack, R.W.: The Effect of Viscosity in Hypervelocity Impact Cratering. AIAA paper 69-354.
- [2] MacCormack, R.W.: Computational Efficiency Achieved by Time Splitting of Finite Difference Operators. AIAA paper 72-154.
- [3] Beam, R.M. and Warming, R.F.: An Implicit Factored Scheme for the Compressible Navier Stokes Equations. AIAA Journal, Vol 16, No. 4, April 1978, pp 393-402.
- [4] Pulliam, T.H. and Steger, J.L.: On Implicit Finite Difference Simulations of Three Dimensional Flows. AIAA 16th Aerospace Sciences Meeting, Huntsville, AL. Paper 78-10, Jan. 16-18, 1978.
- [5] MacCormack, R.W.: A Numerical Method for Solving the Equations of Compressible Viscous Flow. AIAA paper 81-0110.
- [6] Vinokur, M.: Conservation Equations of Gas-dynamics in Curvilinear Coordinate Systems. J. of Computational Physics, Vol 14, Feb. 1974, pp 105-125.
- [7] Steger, J.L.: Implicit Finite-Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries. AIAA Journal, Vol 16, July 1978, pp 679-686.
- [8] Warming, R.F., Beam, R. and Hyett, B.J.: Diagonalization and Simulations Symmetrization of the Gas-Dynamic Matrices. Math. of Comp., Vol. 29, Oct. 1975, pp 1035-1045.
- [9] Hakkinen, R.J., Greber, I., Trilling, L., and Abarbanel, S.S.: The Interaction of an Oblique Shock Wave with a Laminar Boundary Layer. NASA Memo 2-13-59W, 1959.
- [10] Bush, R.H.: Time Accurate Interval Flow Solutions of the Thin Shear Layer Equations. Gas Turbine & Plasma Dynamics Lab., MIY, Report No. 156, Feb. 1981.

Acknowledgment

The research was sponsored by the Office of Naval Research under contract No. N00014-81-K-0024, monitored by Dr. Albert Wood.

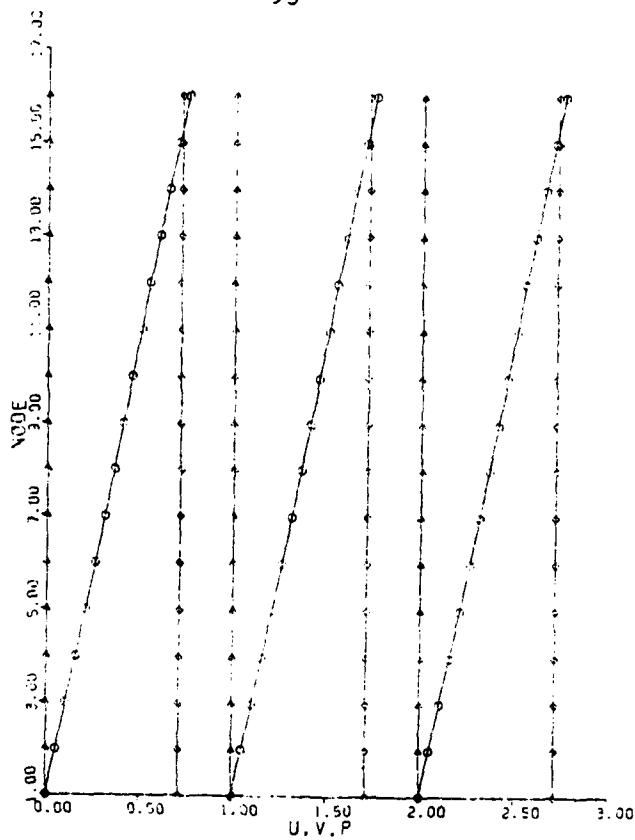


Figure 1. Velocity and pressure profiles for Couette flow at different x locations.

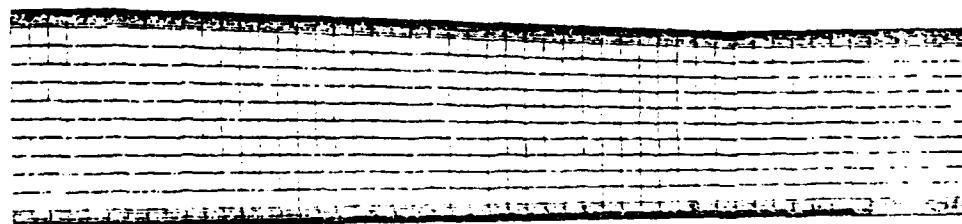


Figure 2. Supersonic diffusor grid.

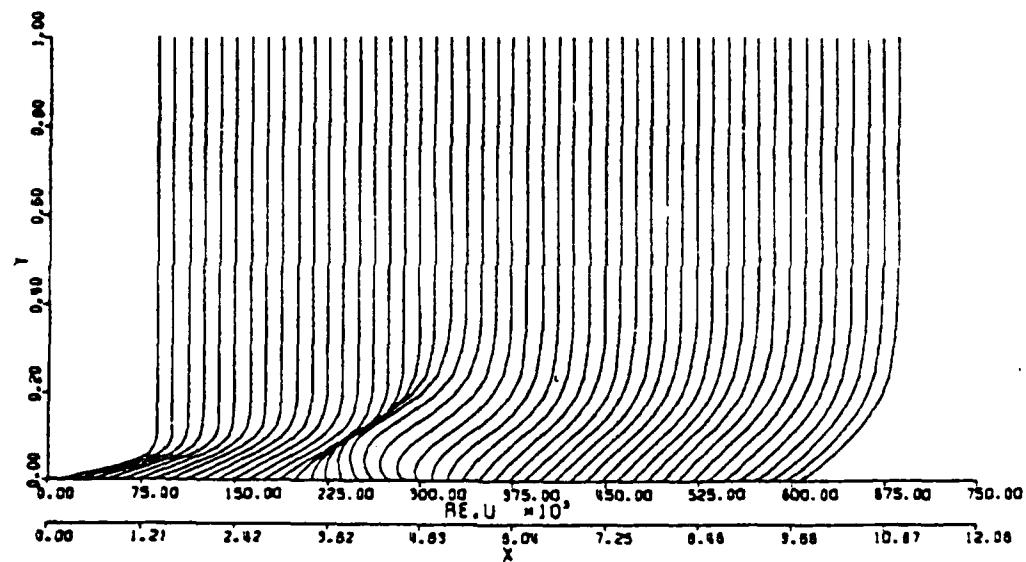


Figure 3. Boundary layer velocity profiles computed for lower wall
shock boundary layer interaction.

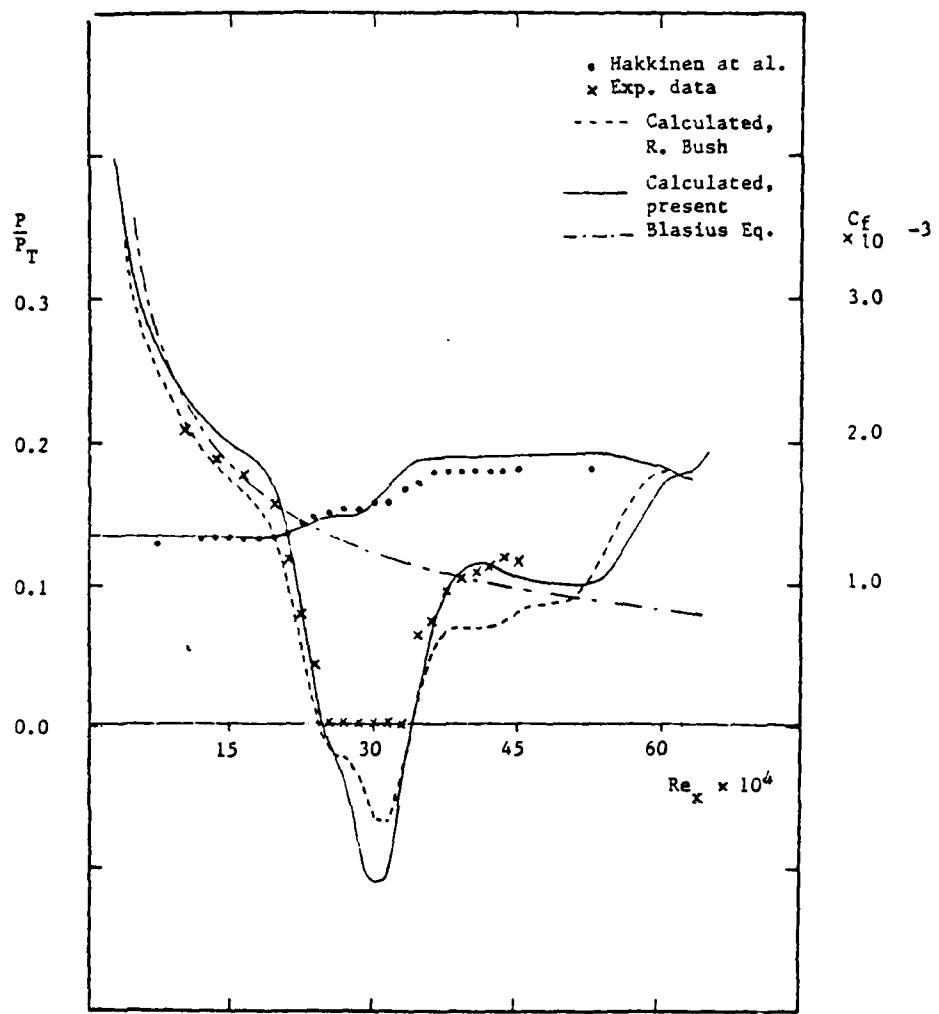


Figure 4. Comparison of lower wall pressure and skin friction distribution with experimental data.

AD-A127 801

EXPERIMENTAL AND THEORETICAL INVESTIGATIONS OF FLOW
FIELDS AND HEAT TRANS. (U) MASSACHUSETTS INST OF TECH
CAMBRIDGE GAS TURBINE AND PLASMA D. E E COVERT ET AL.

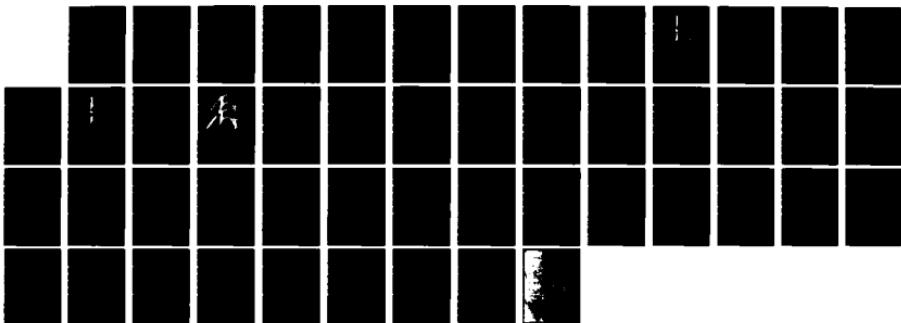
2/2

UNCLASSIFIED

31 DEC 81 N80014-81-K-8024

F/G 28/4

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

IMPLICIT MacCORMACK SCHEME

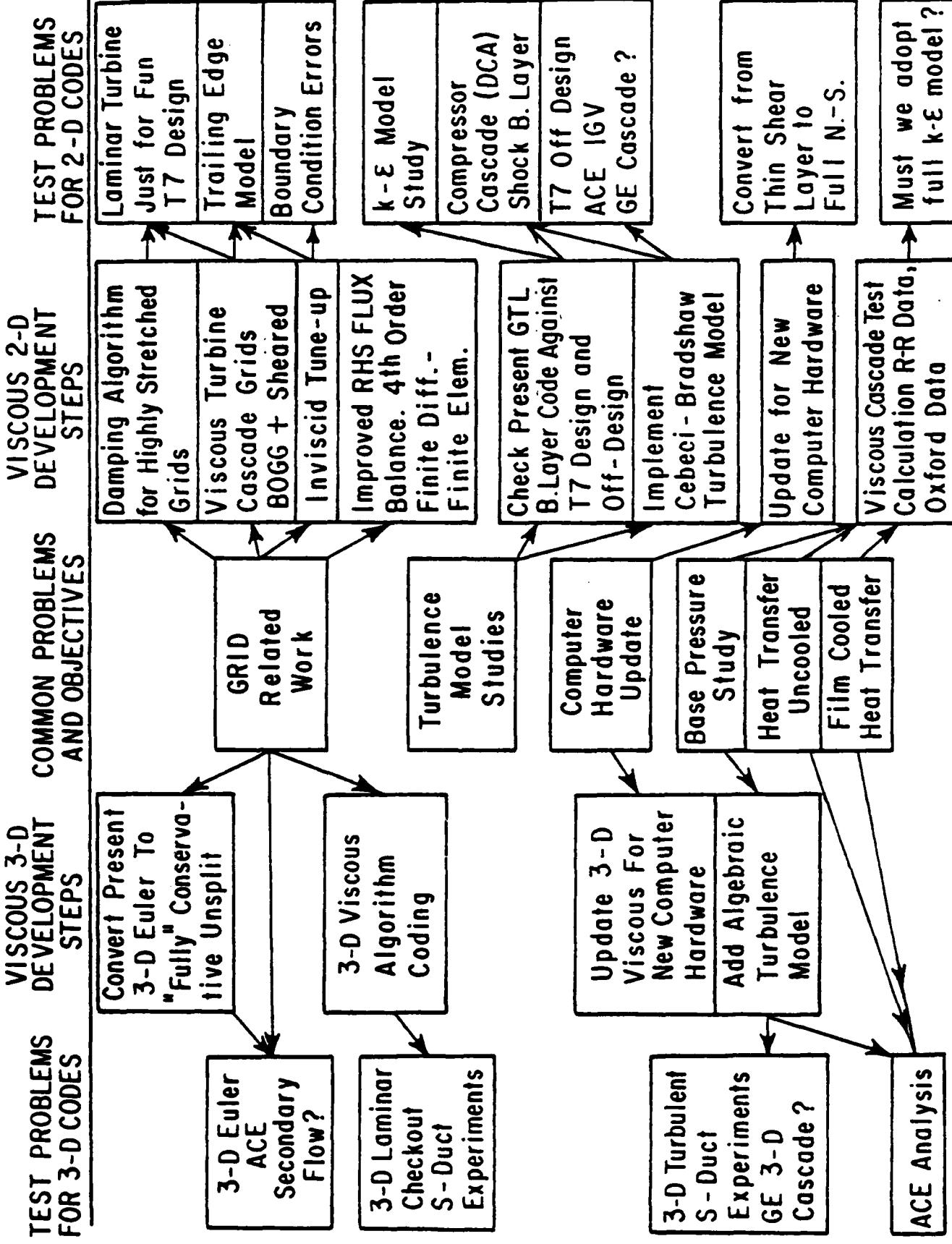
Summary

An extremely interesting implicit scheme for viscous flow calculations was suggested by MacCormack. This scheme was generalized for arbitrary two-dimensional geometries and examined for its use in turbomachinery type geometries. The scheme was found to be potentially useful for supersonic inlet analysis but generally not useful for turbomachinery analysis. A copy of an AIAA paper presented at the 20th Aerospace Sciences meeting is enclosed as part of this report.

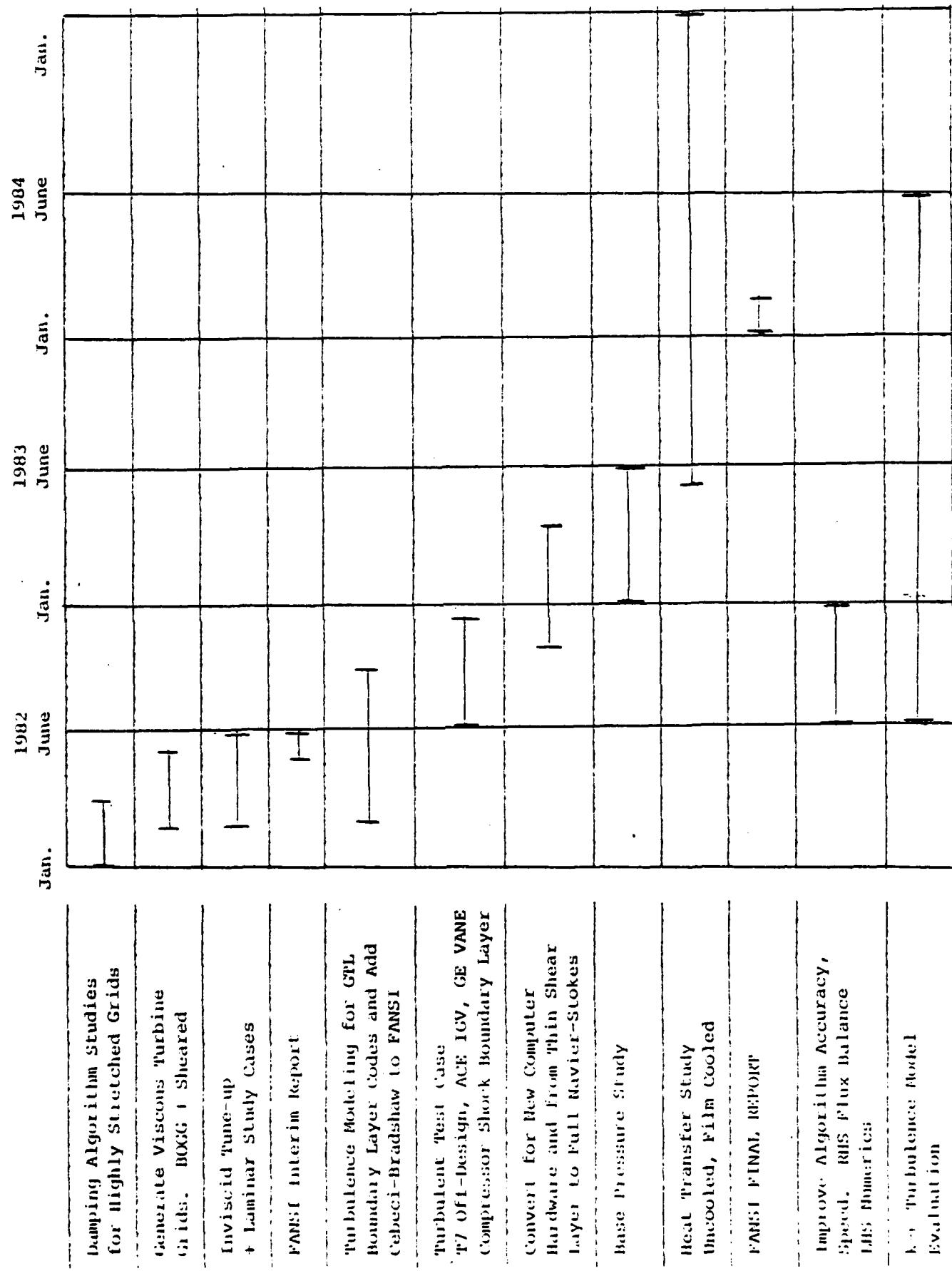
COMPUTATIONAL PLAN

As a program management aid, a three-year computational plan has been created and is outlined in figures (1-3). Figure (1) attempts to outline logical interrelationships between development work on the two and three dimensional viscous codes, specific test examples to be calculated, specific problem areas to be examined and general project objectives. It can be seen that the two-dimensional viscous code is projected to be used both as a development and test tool for the three-dimensional code and to study important flow problems such as base pressure prediction and film cooling heat transfer prediction. Figures 2 and 3 are projected time lines for the code development and checkout. The test examples chosen are expected to be modified through interaction with the sponsors and by events.

LOGIC DIAGRAM FOR VISCOUS CODE DEVELOPMENT



PROJECTED TIMELINE FOR VISCOUS 2-D CODE DEVELOPMENT



PROJECTED TIMELINE FOR VISCOUS 3-D CODE DEVELOPMENT

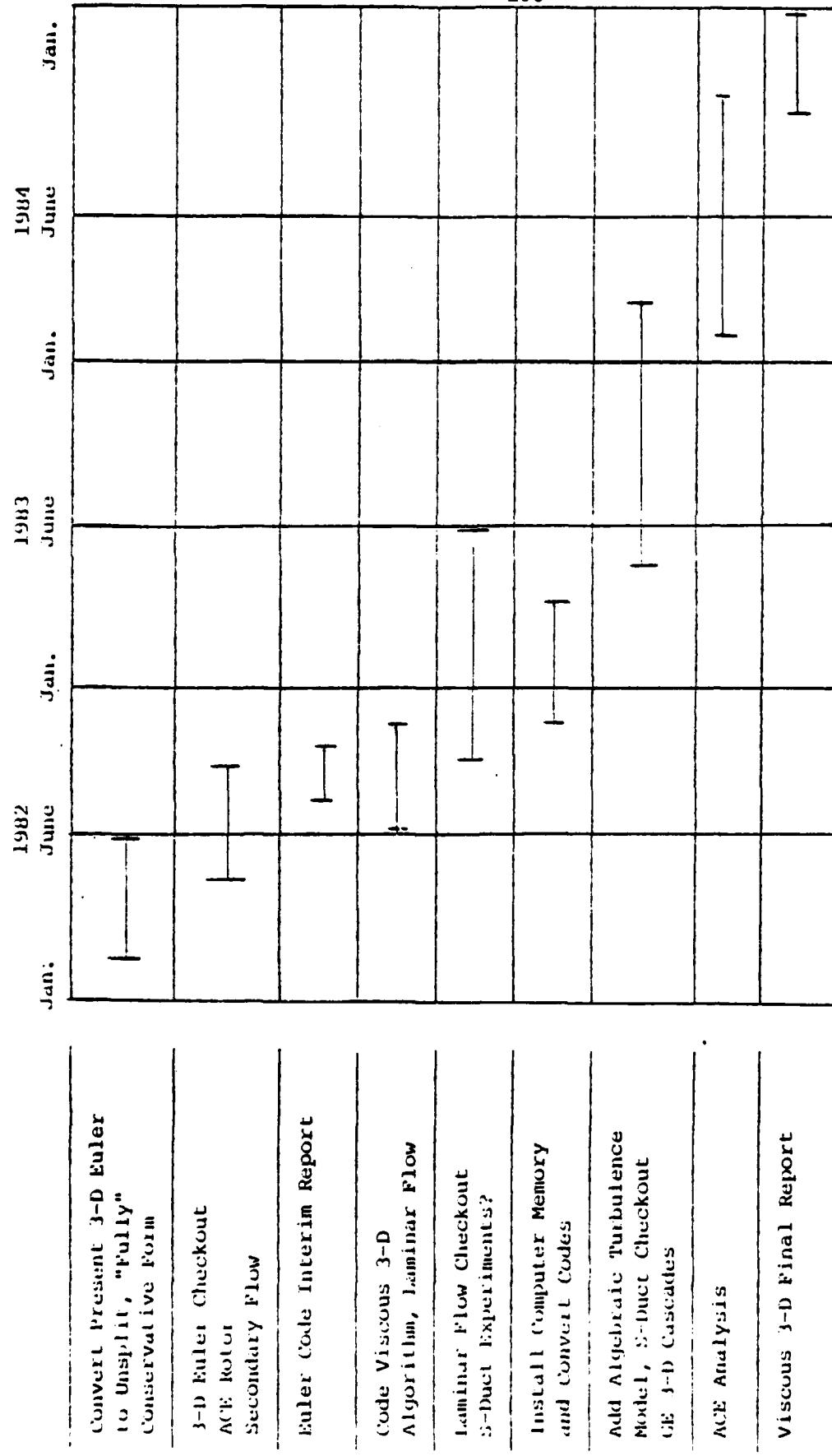


Figure 3

TWO DIMENSIONAL VISCOUS CODE STATUS

Summary

The status of the present two-dimensional viscous code development is illustrated through a series of calculations for the RR T7 cascade at design incidence. The first inviscid calculation illustrates that the problems of global mass and momentum conservation associated with time marching codes have been solved and that adequate blunt leading edge resolution can be obtained with sheared grid systems. Analysis of such solutions has shown that the artificial viscosity terms used to control algorithm stability are themselves an error source when grid spacings change rapidly. Improved damping term formulations are the next major focus for two-dimensional code development. A second set of inviscid and laminar viscous solutions shows the extreme sensitivity of turbine cascade calculations to trailing edge geometry. Since our intent is not to develop an inviscid flow code, inviscid trailing edge model and inviscid boundary condition improvements will be directed by Dr. Norton of R-R.

Inviscid Flow Calculations

A set of test calculations for the RR T7 cascade geometry were conducted in order to evaluate general predictive capabilities of the present code and grid systems. The test calculations were all conducted at design incidence. The first grid system chosen is shown in figure 1 and uses 69 axial points and 30 tangential points. A rather blunt trailing edge with coarse grid resolution was used. Figure 2 shows the comparison between predicted and measured surface Mach numbers. Reasonable agreement exists on the pressure surface. Leading edge resolution appears adequate

but rapid changes near the leading edge appear to be caused by the artificial damping algorithm. Present damping algorithms evolved for grid systems of rather uniform density. When used with the rapidly varying grids of figure 1, this algorithm is inadequate. A stagnation pressure error contour plot for this solution is shown in figure 3. Important pressure errors are generated at the trailing edge, but the solution is generally adequate in this respect. Figures 4 and 5 are plots of the mass averaged flow angle and mass flow rate against axial distance. Conservation of these quantities is excellent.

In order to improve leading and trailing edge resolution a new grid, shown in figure 6, was constructed. This grid has 89 axial points and 50 tangential points. A "sharp" trailing was modeled. The surface Mach number comparison for this grid is shown in figure 7 and can be seen to be very poor. Suction surface velocities and blade lift are very low as is the suction surface trailing edge velocity. This result is quite surprising unless one realizes that the blade lift (circulation) is controlled by the trailing edge model, and the "sharp" trailing edge does not produce a good flow model.

Viscous Flow Calculation

In order to test the importance of trailing edge modeling a laminar, viscous, design incidence solution was computed. The grid, shown in figure 8, has 100 axial points and 50 tangential points. Since the object was to test trailing edge models, no attempt to generate a good viscous grid was made. Instead, the same leading and trailing edge axial spacing was retained, and 10 points added near each blade surface. Computed surface

pressures are shown in figure 9 and can be seen to compare much more favorably with the measured pressures than do the second inviscid test case. The same type of leading edge pressure oscillations can also be seen in this solution.

Implication for Viscous Code Development

The proper conclusion to be drawn from these comparisons is not that viscous effects are important for the T7 at design incidence, but rather that trailing edge modeling is important for inviscid calculations. Since our objective is not really to develop inviscid code predictions, trailing edge models and boundary condition procedures for inviscid calculation will be directed by Dr. Norton of RR. Since boundary layer calculations and turbulence modeling are also being followed by Dr. Norton, the possible use of these codes for inviscid predictions will be adequately explored.

These examples demonstrated persistent leading edge oscillations which are felt to be not really due to blunt leading edges but an artifact of the artificial damping algorithm when used with rapidly varying grid mesh spacings. This problem and the generation of more realistic viscous grids are to be the next areas of work for the two-dimensional code.

T7 TEST CASE, INVISCID,
SHARP TRAILING EDGE

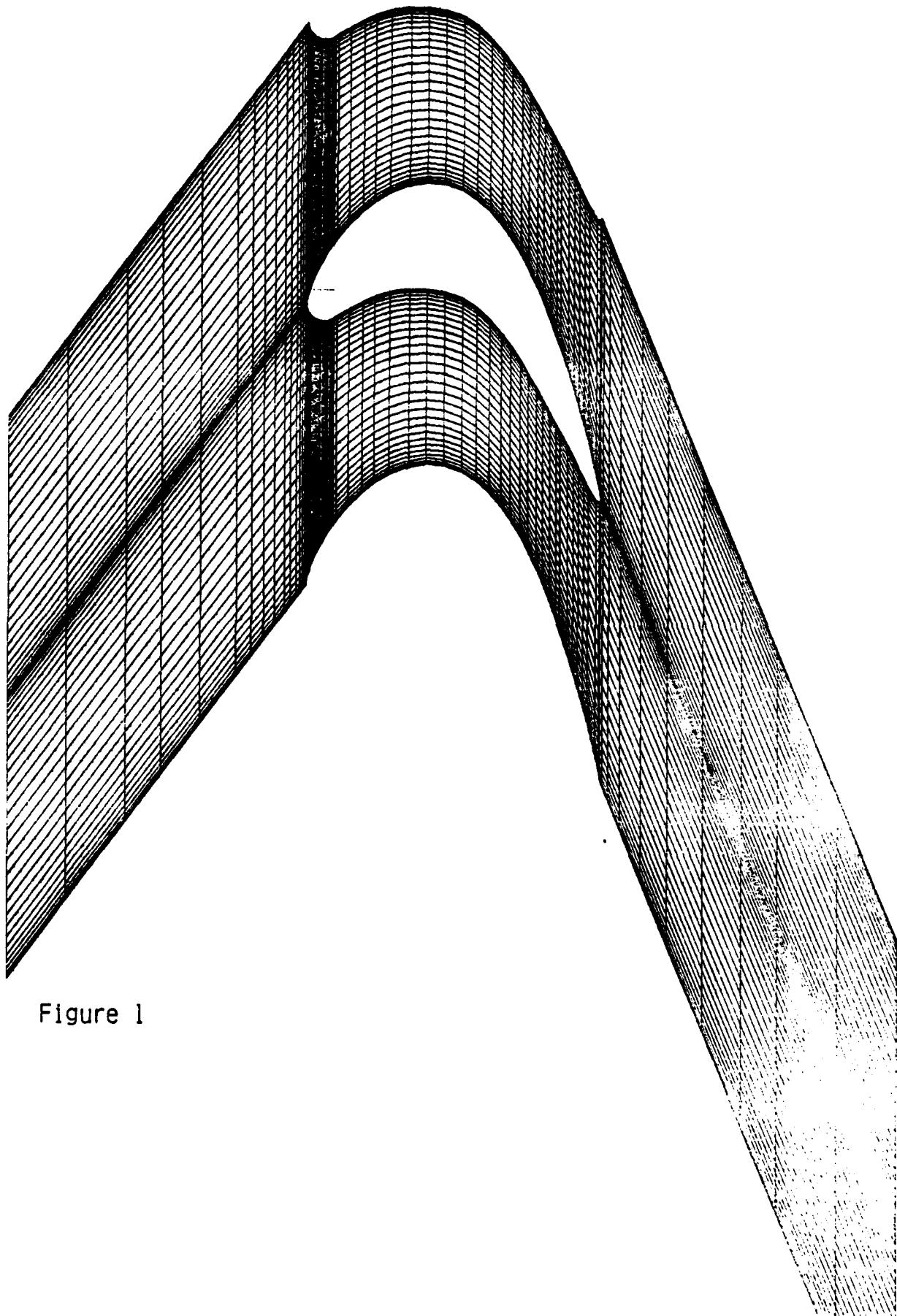


Figure 1

T7 TEST CASE, INVISCID CALCULATION, BLUNT TRAILING EDGE

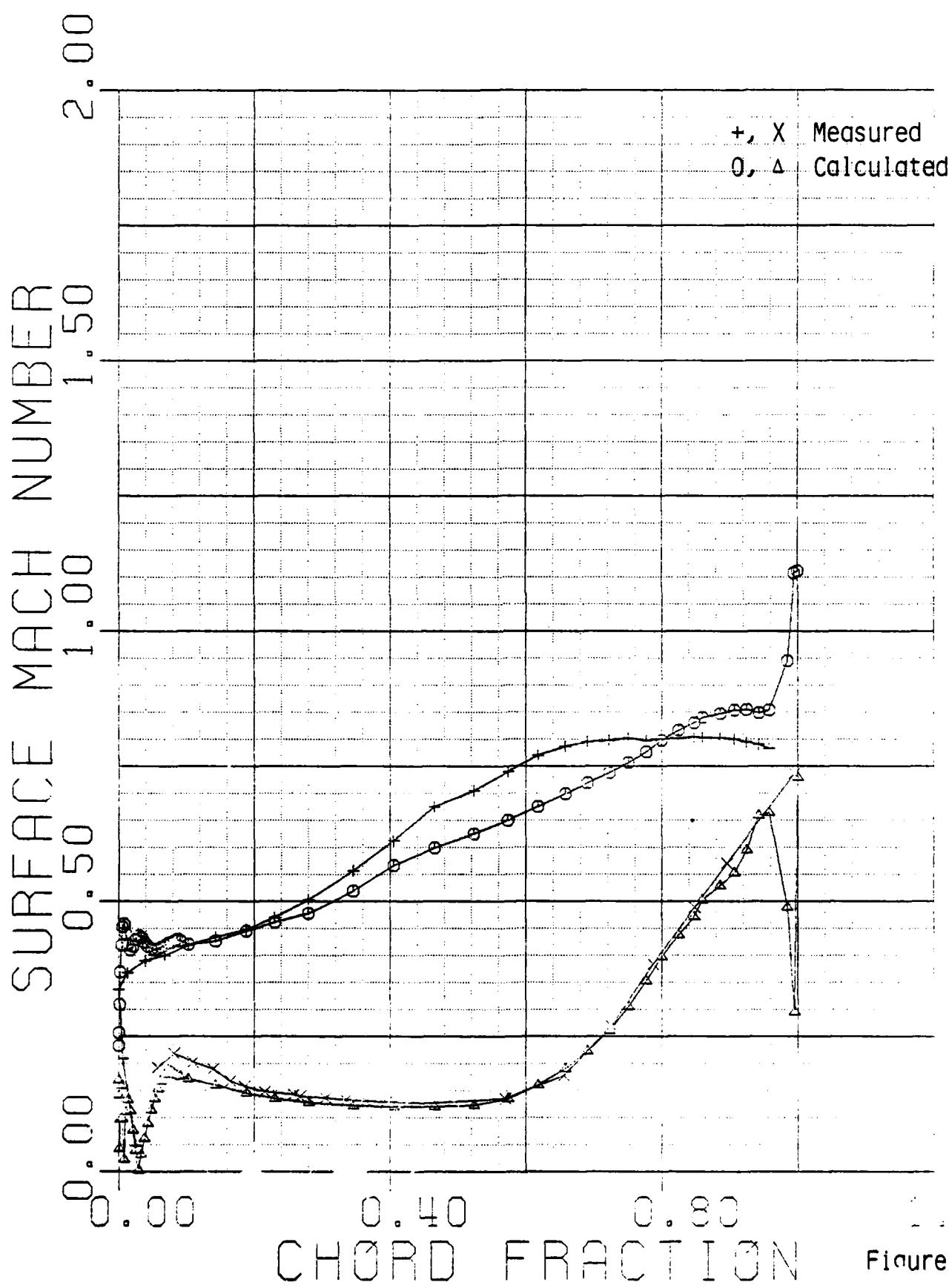
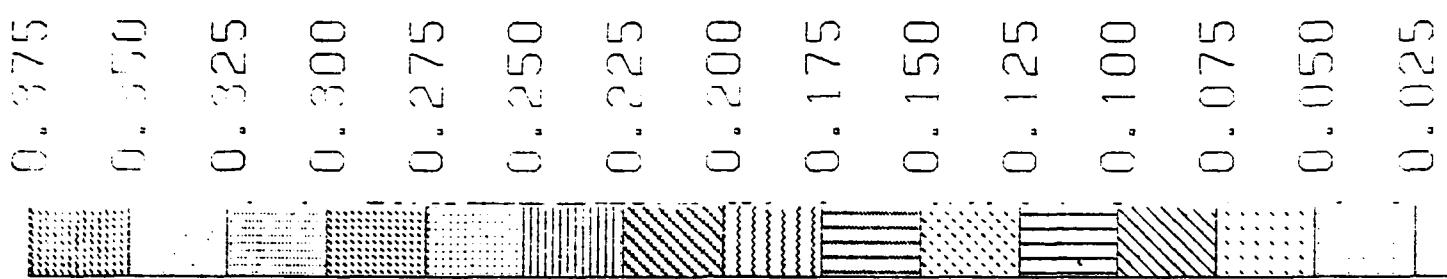


Figure 2



T7 TEST CASE
STAGNATION PRESSURE ERROR

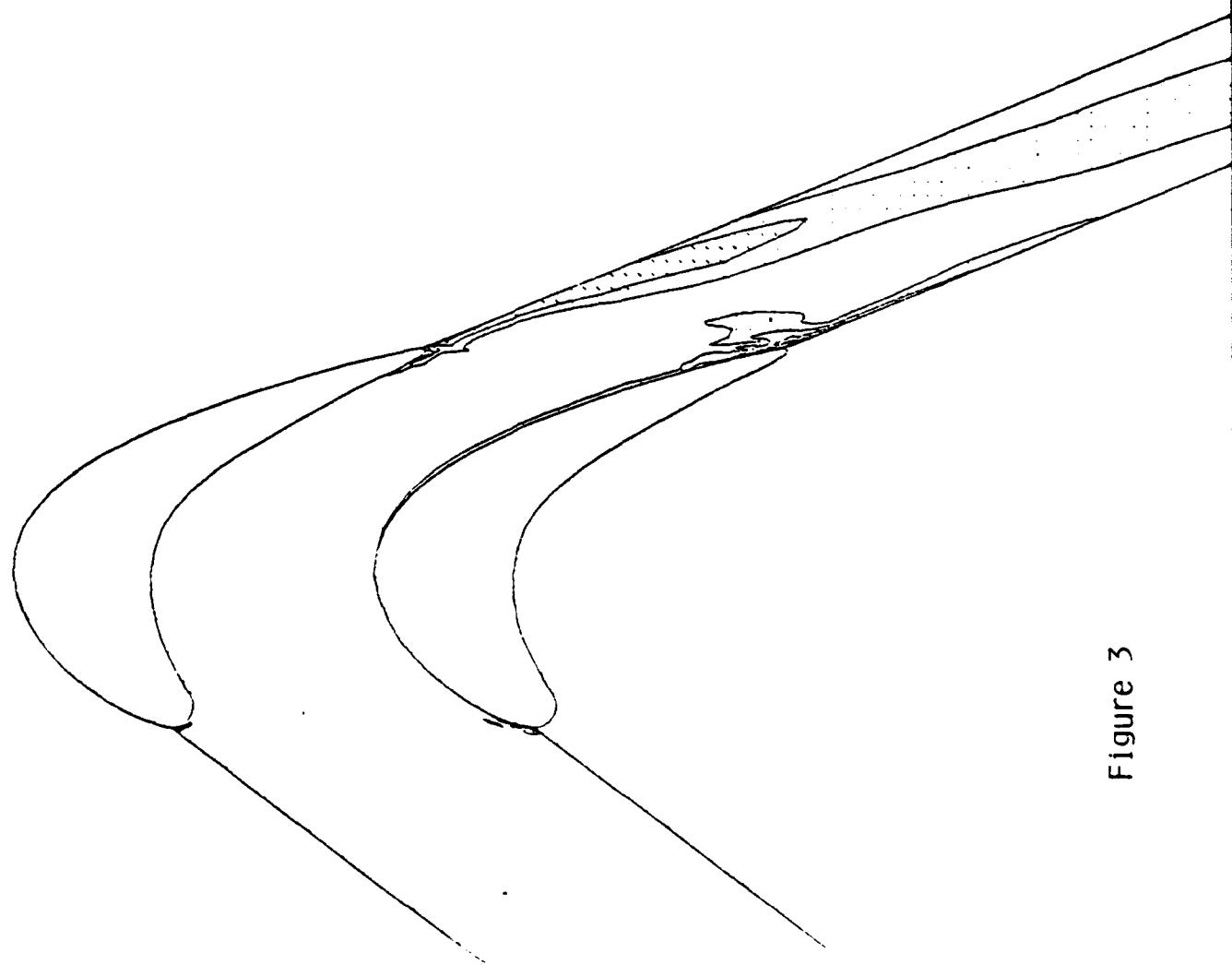


Figure 3

T7 TEST CASE

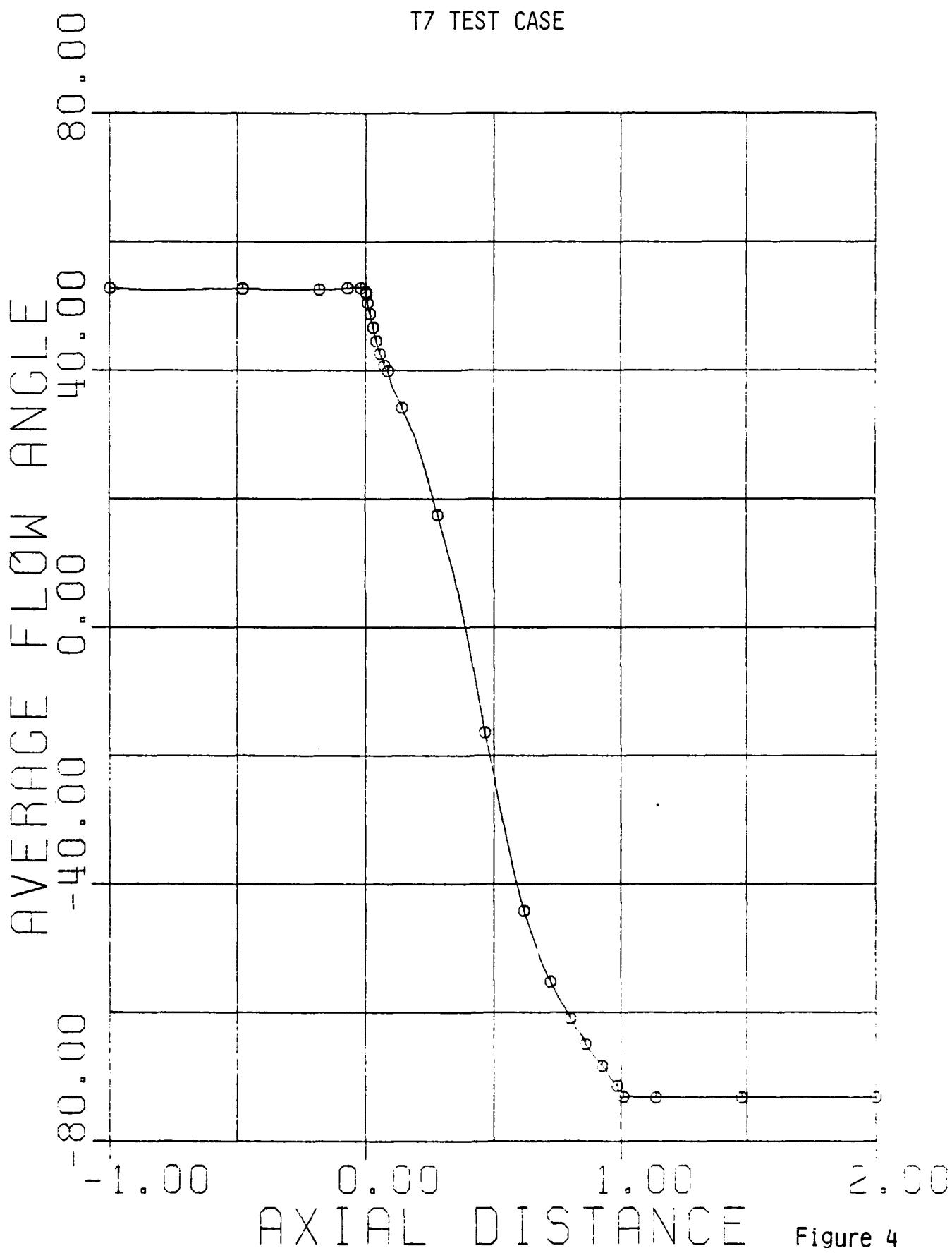


Figure 4

T7 TEST CASE

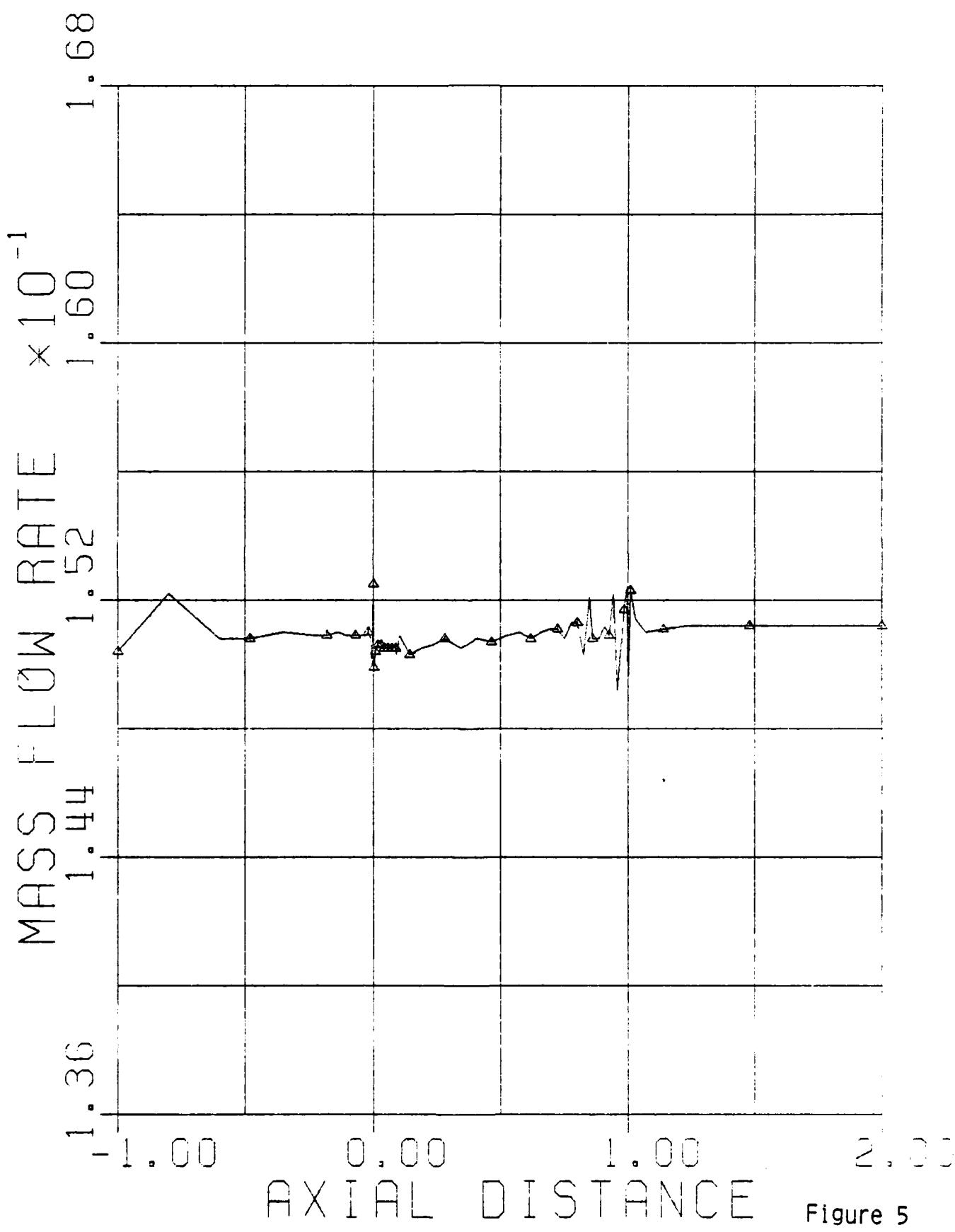


Figure 5

¹⁰⁹
T7 TEST CASE, INVISCID,
SHARP TRAILING EDGE

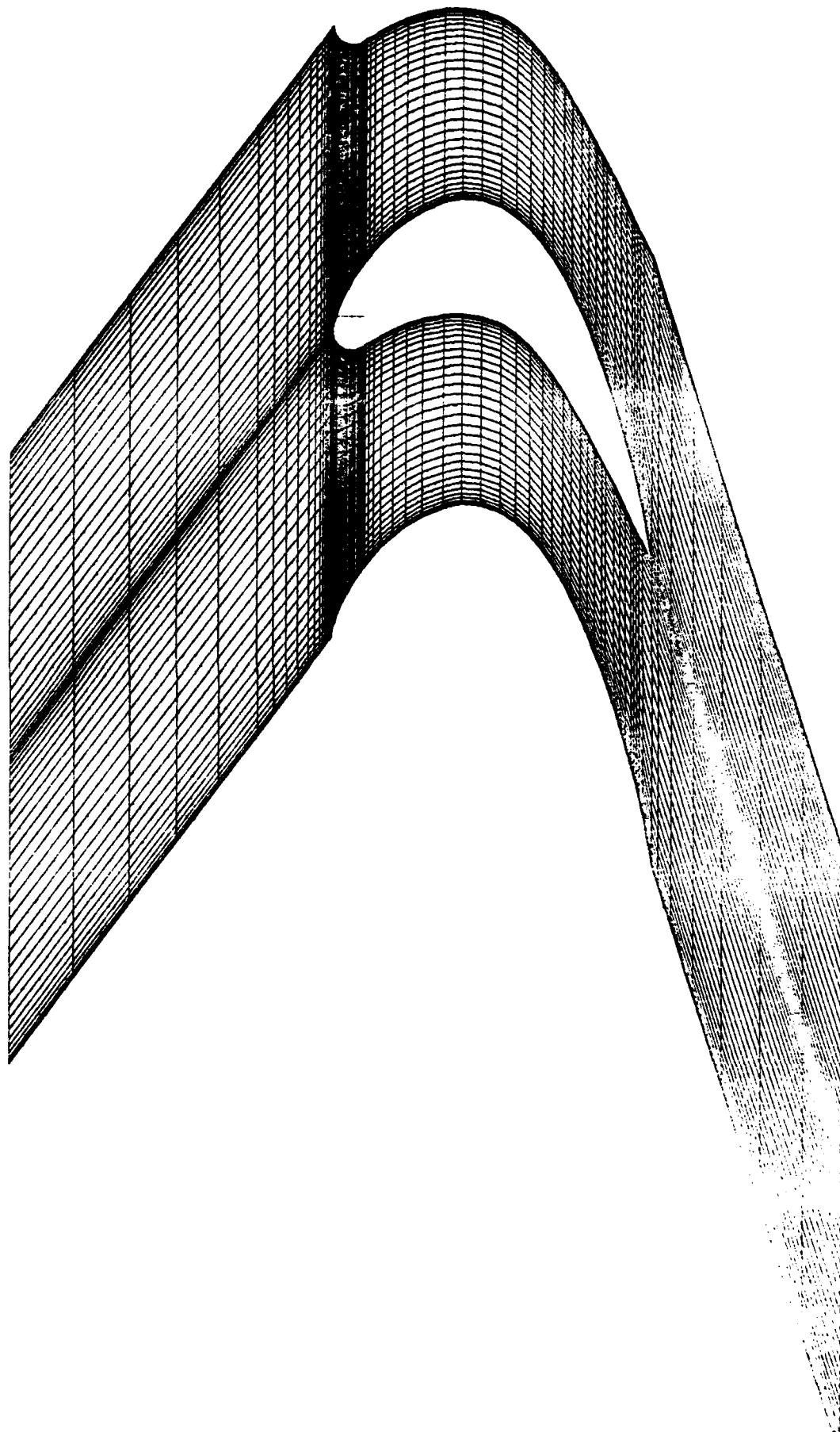


Figure 6

T7 TEST CASE, INVISCID CALCULATION, SHARP TRAILING EDGE

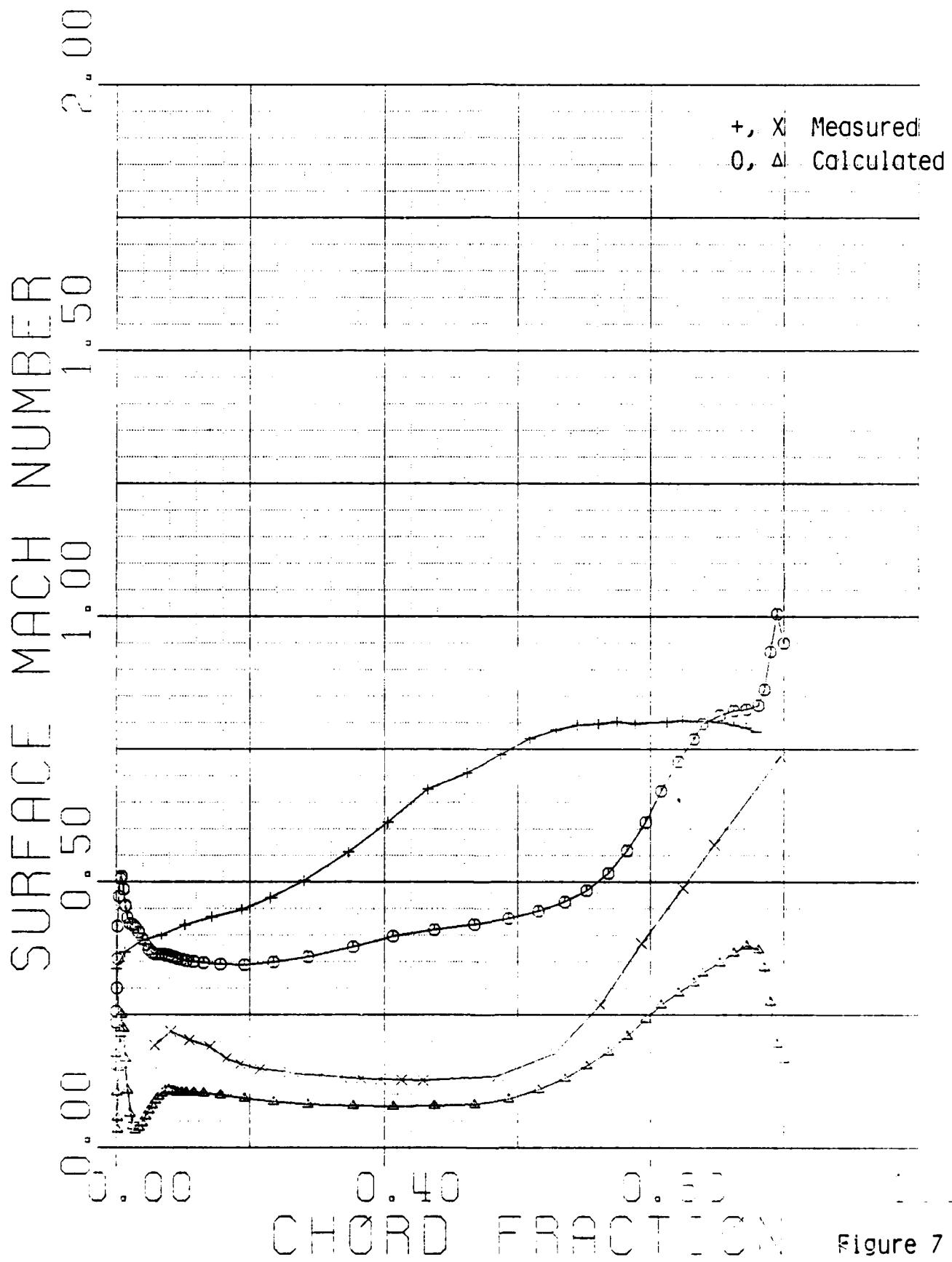


Figure 7

111
T7 TEST CASE, VISCOUS & GRID

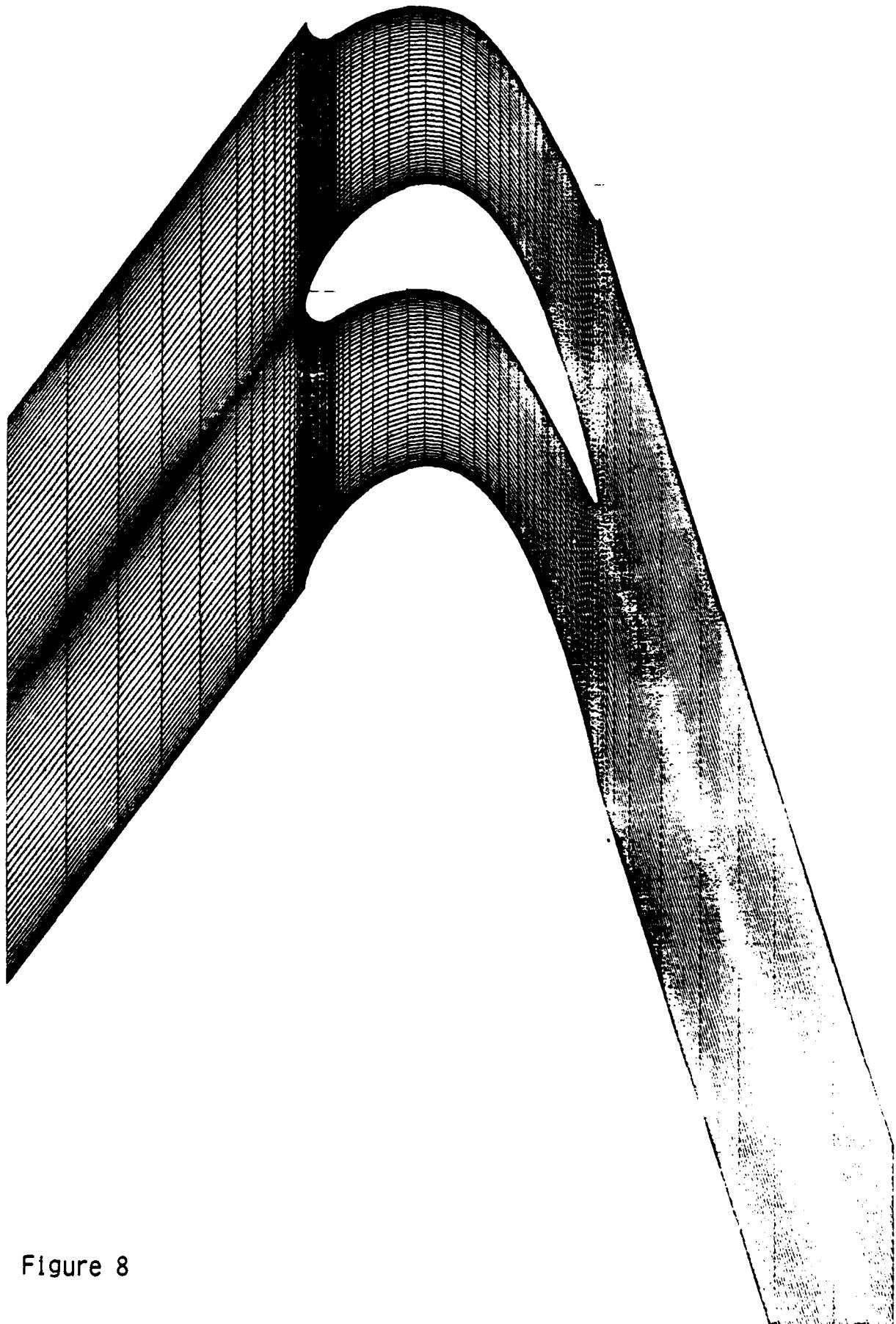


Figure 8

T7 TEST CASE, LAMINAR CALCULATION, BLUNT TRAILING EDGE

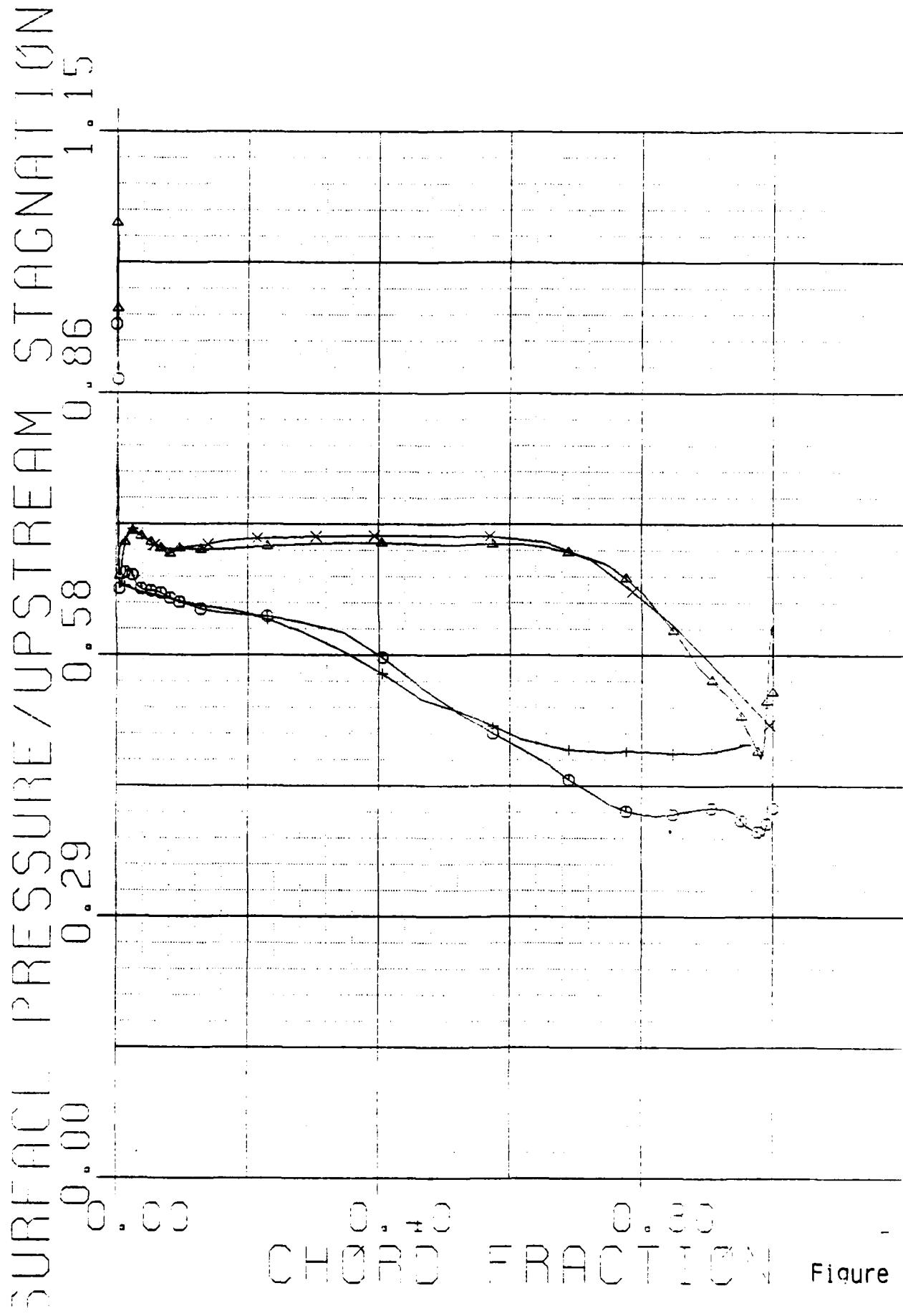


Figure 9

FIRST ANNUAL RRI/ONR REPORT

EXPERIMENTAL EFFORTS

Introduction

This report summarizes the experimental activities during the first year of the Investigations of Flow Fields and Heat Transfer in Modern Gas Turbines. Activities during this period consisted primarily of the conceptual and preliminary mechanical design of the Blowdown Turbine Facility.

The goals of the program are summarized in Figure 1. The emphasis is on developing the capability to simultaneously measure the heat transfer and aerodynamics of a full size film cooled high pressure turbine under rigorously simulated conditions. Thus, all the nondimensional force and energy ratios (Reynolds No., Mach No., Rossby No., Prandtl No., Eckert No.) will be kept the same as for the full scale turbine. This must be accomplished, however, at a cost level, both for construction and operation, consistent with a University operation.

Scaling

The principal scaling is one of temperature. The temperature ratios, main flow gas to metal and coolant flow to metal, are kept the same as for the full scale turbine but the absolute temperature levels are reduced. In this case, the metal temperature is scaled to room temperature. The resultant gas temperatures are 478°K for the main flow and 212°K for the coolant.

Another variable in the scaling is that of gas composition. Air is

a possible choice but has a ratio of specific heats, $\gamma = 1.38$ at 478°K instead of the $\gamma = 1.28$ typical of high temperature turbines. By mixing light and heavy gases we can adjust γ , both to simulate the full scale and to investigate the importance of γ matching for aerodynamics & heat transfer testing. Figure 2 lists the properties of several argon-refrigerant mixtures while Table 3 lists the inlet conditions required for similarity with the full scale turbine. Note that not only does the gas mixtures match γ , but the Reynolds No. is matched at a much lower pressure, 4-5 atm., implying considerable savings in tanks, piping, etc. as well as in the amount of test gas required. Of the refrigerants investigated, R-12 (Freon-12) is by far the least expensive (\$1/lb) and therefore the gas of choice for the main flow. The coolant temperature is low enough to condense R-12 at the max pressure considered, 10 atm, so that R-14 would be used for the coolant. Other advantages of the gas mixture stem from the 25% reduction in the speed of sound compared to that in air at the same temperature. Thus, for a given tip Mach No., the rotational speed is only 3/4 as great-reducing stresses by a factor of 2. The blade passing frequency is similarly reduced, decreasing the frequency response required of the fixed frame instrumentation for a particular spatial resolution in the rotor frame.

Figure 4 illustrates the Blowdown Turbine scaling. Note that the temperature ratios (the Eckert No.) establishes the gas temperatures; the Reynolds number establishes the inlet pressure; the ratio of specific heats establishes the gas composition; and the tip Mach No. sets the rotational speed. The Prandtl No. cannot be independently set

but fortuitously is essentially exactly correct. The reduced absolute pressure levels and the high molecular weight of the mixture combine to reduce the turbine power produced by a factor of approximately 20.

At this point, the turbine size must be selected. Basically, as large a turbine as possible is desirable from the point of view of instrumentation. In fact, even the largest production high pressure turbines typically have blade spans of only 6 cm. The facility size scales more closely with mass flow than blade length, however. The mass flow required for a turbine from a 50,000 lb thrust engine was beyond the resources of this program and a 0.5 m (20 inch) diameter turbine size was chosen. This was consistent with the resources in hand, comparable in size to the NASA High Pressure facility, and has a span 2/3 of that of the largest turbines.

Steady state turbine performance measurement is not a strong driver since the goals of this program are primarily concerned with time resolved measurements. Therefore, the idea of building the facility around an existing conventionally tested turbine had considerable attractiveness. It reduced the program costs and, even more importantly, provides a baseline performance against which to compare the blowdown test results. The turbine selected is of Rolls Royce Limited design and manufacturer. It represents a high work design of the mid to late 1970's and has a pressure ratio of approximately 4 to 1. The full scale column in Figure 4 refers to this turbine.

For purposes of future research capabilities the facility was sized for twice full scale Reynolds No. testing (10 atmos. laboratory inlet total

pressure) and a laboratory inlet temperature of 530°K (500°F), to permit carbon steel construction and the use of elastomer seals. This results in a facility capable of simulating full scale conditions up to 40 atmos. at 2500°K (4500°R) as shown in Figure 5.

Facility Configuration

Now that the physical scaling is established, the facility configuration must be selected. Many candidate configurations were examined with four in some detail. These are summarized in Figure 6. All of these designs are based on transient testing. This is both to reduce the costs associated with the experiment and to take advantage of many of the transient heat transfer and fluid dynamic testing techniques developed over the last 25 years. Also, since the predominant source of unsteadiness in turbines is rotor-stator interactions, a test time on the order of 0.1 to 1.0 sec. (i.e. 600 to 6000 blade passings) should be sufficient for most studies. This has proved true in several transient cascade, turbine, and compressor facilities at MIT and around the world.

One configuration studied used room temperature high pressure gas storage (100 to 200 atmos.) to feed a flow heater and then the turbine. A large flywheel would absorb the turbine power (Figure 5, #1). This has the advantage of maintaining constant inlet pressure during the test time with fast acting pressure regulators. The principal disadvantage is that the high pressure storage precludes the use of a heavy gas mixture (the heavy gases condense at high pressure).

The second configuration is a variation on the first in which

the high pressure storage is replaced by a room pressure expulsion bladder contained in a heated shell. Since, the turbine inlet pressure is constant at room pressure, Freon can be used. This scheme has the disadvantage that full Reynolds number similarly cannot be achieved. Also, the technology required for a 100 m³ 500°K balloon is risky.

The third configuration considered was the original concept which motivated this project. In this scheme the turbine is directly coupled to a compressor whose inlet is the turbine's outlet. Thus, the turbine and compressor turn at the same mechanical speed and share the same flowpath. Since a choked turbine produces power proportional to the tip Mach No. squared and a centrifugal compressor does work as the square of the tip Mach No., the turbine-compressor pair are matched over a considerable range and operate at constant corrected speed without the need of a control system. The corrected weight flow is constant so long as any orifice in the flow path remains choked. If we now place the turbine-compressor test section between two tanks, the upstream supply tank filled with the gas mixture at appropriate conditions and the downstream dump tank evacuated to full vacuum, the facility is complete with the addition of a valve or diaphragm between the supply tank and test section.

The design of such a blowdown tunnel was carried out in some detail. A compressor was chosen in this design over a flywheel for two reasons. The first is that the compressor can maintain a closer match in corrected speed compared to the flywheel since with the flywheel, the

turbine mechanical speed must always increase (if only a small amount), while with the compressor, the turbine speed can slow to match the drop in inlet temperature that is inherent to a blowdown from a fixed volume tank. Thus, the flywheel may be appropriate for a constant inlet condition scheme such as Figure 5, Nos. 1 & 2, but an energy absorber is really required for a blowdown. The compressor design proved to be a problem, however. The problem with the centrifugal compressor is that the very high work output of the transonic turbine requires a very large diameter compressor wheel which in itself acts as a large flywheel. As the compressor diameter passed 1 meter in the design, it was clear that a safe mechanical design of the rotating system would be very difficult. (Note that the aerodynamic design is not so difficult since the compressor efficiency does not effect its power absorption, only the pressure rise. The pressure rise does, however, determine the volume of the dump tank required. The principle aerodynamic design consideration for the compressor is that the weight flow vs. speed characteristic match that of the turbine).

Various compressor designs were then attempted including a supersonic axial compressor and the adaptation of a multistage aircraft engine high compressor. These schemes did not seem to offer much in reducing either the mechanical complexity or the technical risk of the project.

Another configuration investigated replaced the compressor with an eddy current brake. An eddy brake consists of a moving conductor (metal disk or drum) in an imposed magnetic field. This induces

secondary currents (eddy currents) in the conductor which opposes the applied field, thus generating a braking force. The power is dissipated as heat in the conductor. The eddy brake is attractive because the braking torque is proportional to the square of the rotational speed. Thus, an eddy brake coupled to the turbine will, as the compressor, match the turbine characteristic without the need of a control system. The eddy brake offered the advantage of a high power density design-simplifying packaging and rig layout. Unlike a mechanical friction brake, the eddy brake has no moving parts other than the conductor and thus relatively few mechanical design problems.

Due to the above potential advantages of the eddy current brake and also due to the investigators expertise in compressor design and ignorance of eddy current brakes, the eddy brake configuration was chosen for the Blowdown Turbine.

Given the selected configuration, and a 0.2 to 0.4 sec. run time goal, various facility sizes were simulated. Figure 8 illustrates that for a typical eddy brake configuration facility, the corrected speed and turbine pressure ratio remain constant to 1% over 0.4 seconds.

Subsystem Detail

Main Valve

A principle component of this facility is the main valve which separates the supply tank from the test section. The valve must seal vacuum against 10 atmospheres pressure at 500°K (500°F), open fully in 50 ms, and provide a smooth disturbance free inflow to the turbine (Figure 9). The explosively ruptured diaphragm that was used in the MIT

Blowdown Compressor Facility could not be used here because of the higher pressure and temperature. The design chosen was the pilot operated annular plug valve illustrated in Figure 10. The valve is constructed of steel and is therefore fairly massive (slider weight = 100 Kg). It is designed to operate in thermal equilibrium and is thus oil heated. Valve dynamics depend upon a 100 mm diameter pilot pneumatic piston for the initial motion, with the major part of the opening force coming from the pressure difference between the outside surface of the slider and the inside damping chamber which is initially evacuated. As the slider moves, the damping chamber fills through the orifice flow path and then acts as the damper, slowing the slider to a stop. To aid in the design, the valve dynamics were simulated on a computer. Typical predictions from the simulation are presented in Figure 11. The valve is at its designated open position (i.e. valve flow area is greater than any downstream flow area) after less than 50 ms. from initial motion point.

Eddy Brake

The design of the eddy brake has proven to be a considerable technical challenge. An eddy brake torque vs. speed characteristic is shown in Figure 12. The Blowdown Turbine must operate on the linear section of the curve if the power absorbed is to be proportional to the square of the shaft speed. As the shaft speed increases, the magnitude of the eddy currents increase until at ω_0 , the eddy field is equal to the applied field. The torque will now decrease with increasing speed as the eddy field excludes the applied field flux lines.

The challenge is to design a brake with a critical speed, ω_0 , above the turbine operating range but which is still capable of absorbing the requisite power in a reasonable mechanical package. To simplify the problem, the brake was designed to heat sink the power produced (1.2 MW for 1 sec.) in the moving conductor. This introduced an additional design problem since at a given mechanical speed, ω_0 decreases with increasing conductor thickness. Thus the conductor must be thick to absorb the power but it must be thin because of the electrodynamics. The solution adopted was to use a very high resistivity, high hot strength material in a drum configuration with 10 "horseshoe" electromagnets arranged around the periphery. The design is summarized in Figure 13. Current from a D.C. source is supplied to the magnets as the valve is open (the rotor is to already have been brought to operating speed by a small D.C. motor).

Tanks and Auxiliary Systems

The supply and dump tanks are of carbon steel construction. The supply tank volume is approximately 14 m^3 (365 cu. ft.) and dump tank volume is 20 m^3 (550 cu. ft.). The supply tank is double walled so that hot oil can be circulated to heat the tank to the desired operating point. The valve is similarly heated. A silicon rubber insert and a water jacket on the upstream flange keep the test section at room temperature. A commercial 80 KW electric oil system heats and circulates the oil. The facility is illustrated in Figure 14.

Other auxiliary systems include a vacuum pump for evacuating the tunnel and a gas mixture system required to supply the mixture. The

facility is designed with a 2 to 3 hour recycle time.

Data Acquisition

The data acquisition and analysis system for this facility must be capable of acquiring data with an information bandwidth of at least 40 kHz from 20 to 100 sensors for at least 0.5 seconds. This implies a prodigious data rate and volume. No commercial system was capable of meeting these requirements at a reasonable price. The preliminary design of a custom system was performed and its detailed design and construction contracted out. The system is shown schematically in Figure 15. It consists of up to 100 separate channels each capable of digitizing at 200,000 samples/sec. There are also up to 96 lower speed channels multiplexed onto 8 high speed channels. The digital output from these channels is transmitted 200 ft. at an average data rate of 256 Mb/sec. to a 32 M byte semiconductor memory, thus permitting up to 200,000 data samples on each of 100 channels. The data acquisition system is under the control of a local microcomputer. After the test is completed, the data is read from the memory into a Perkin-Elmer 3242 computer for analysis. System specifications are summarized in Figure 16. Initially, 25 high speed and 48 low speed channels have been ordered.

InstrumentationFlowfield

One of the principle advantages of the scaled blowdown scheme is that the environment is quite benign when compared with a full scale turbine. It is in many ways comparable with that of a conventional compressor and thus most of the standard compressor instrumentation techniques can be directly applied to the blowdown turbine. The basic high frequency technology is the semiconductor diaphragm pressure transducer such as those manufactured by Kulite, Entran, or Endevco. These are incorporated into various probe configurations with net frequency response in the 10 to 20 kHz range (Figure 17).

Heat Flux

Heat frequency heat flux measurements on the rotor blades are the first goal of this program. Several alternate techniques have been considered as summarized in Figure 18. The most promising technique involves the measurement of the temperature difference across a well calibrated insulator as shown in Figure 19. Considerable effort will be spent in developing these techniques.

Conclusions

A Blowdown Turbine Facility has been designed to be capable of accurate simulation of the fluid physics of high pressure film cooled turbines. Detailed design and construction of this device will proceed along with development of the required instrumentation techniques during the next year.

FIGUPE 1

BLOWDOWN TURBINE PRIMARY GOALS

- FULL FLUID PHYSICS SIMULATION - RE, M, PR, RO, $\frac{\Delta T}{T}$
- TIME RESOLVED MEASUREMENTS
 - HEAT TRANSFER
 - AERODYNAMICS
 - SIMULTANEOUS HT & AERO
 - COOLING AIR TRANSPORT
- LOW CONSTRUCTION COSTS
- LOW OPERATING COSTS - UNIVERSITY SCALE

FIGURE 2

ARGON - REFRIGERANT MIXTURES

COMPOSITION AND THERMOPHYSICAL PROPERTIES

$$\gamma_M = 1.28, \quad T = 400^\circ F, \quad P = 14.696 \text{ psia}$$

Refrigerant		Argon - Refrigerant Mixtures				
Mass%	Mole%	$\frac{M_M}{1\text{lbm}}$	$\frac{C_{PM}}{\text{BTU/lbm-}^\circ\text{R}}$	$\frac{R_M}{\text{BTU/lbm-}^\circ\text{R}}$	γ_M	$\frac{k_M}{(\frac{1\text{lbm}}{\text{sec-ft}})}$
R-12	50.94	25.54	60.627	.1500	3.276×10^{-2}	1.279
R-13	49.58	27.32	57.575	.1579	3.449×10^{-2}	1.279
R-13B1	57.84	26.90	69.256	.1313	2.867×10^{-2}	1.279
R-14	47.80	29.36	54.057	.1682	3.674×10^{-2}	1.279
R-22	52.55	33.84	55.692	.1633	3.566×10^{-2}	1.279
R-115	39.97	14.69	56.765	.1602	3.498×10^{-2}	1.279
Air at $1780^\circ K$, 14.696 psia		28.966	$.3066$	6.856×10^{-2}	1.289	3.892×10^{-5}
						15.86×10^{-6}
						.752

FIGURE 3

ARGON-REFRIGERANT MIXTURES: REYNOLDS SIMILARITY

TEST PRESSURES AND SATURATION TEMPERATURES

<u>MIXTURE</u>		<u>CONDITIONS AT REYNOLDS SIMILARITY</u>			
	x_R	P'_O/P_O	P_O (psia)	P_R (psia)	T_{sat} (°F)
R-12	.2554	.2229	64.3	16.4	-17
R-13					
R-13B1	.2690	.2265	65.3	17.6	-65
R-14	.2936	.2637	76.1	22.3	-187
R-22	.3384	.2294	66.2	22.4	-24
R-115	.1469	.2484	71.7	10.5	-50

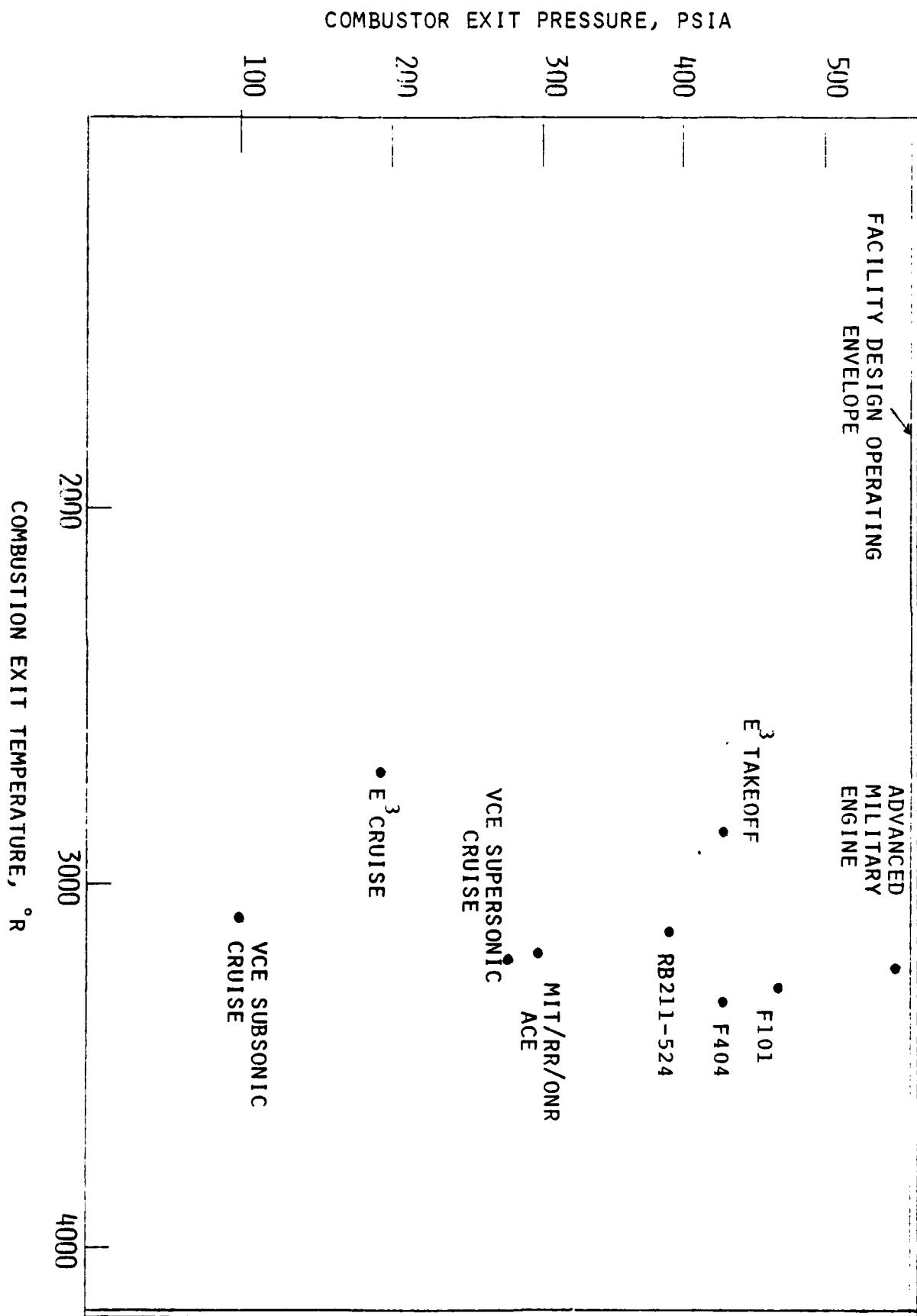
FIGURE 4

MIT BLOWDOWN TURBINE SCALING

	<u>FULL SCALE</u>	<u>MIT BLOWDOWN</u>
FLUID	AIR	AR - R12
RATIO SPECIFIC HEATS	1.27	1.27
METAL/GAS TEMP RATIO T_M/T_G	.63	.63
MEAN METAL TEMP, T_M	1118°K	300°K
INLET TOTAL TEMP, T_G	1780°K	478°K
COOLING AIR TEMP	790°K	212°K
AIRFOIL COOLING AIR FLOW	12.5%	12.5%
TRUE NGV CHORD	8.0CM	5.9CM
REYNOLDS NUMBER	2.7×10^6	2.7×10^6
INLET TOTAL PRESSURE	289 PSIA	64 PSIA
PRANDTL NUMBER	.752	.755
TEST TIME	CONT.	.2 SEC
POWER PRODUCED	24 MW	1.3 MW

FIGURE 5

BLOWDOWN TURBINE SIMULATION CAPABILITIES



CANDIDATE FACILITY CONFIGURATIONS

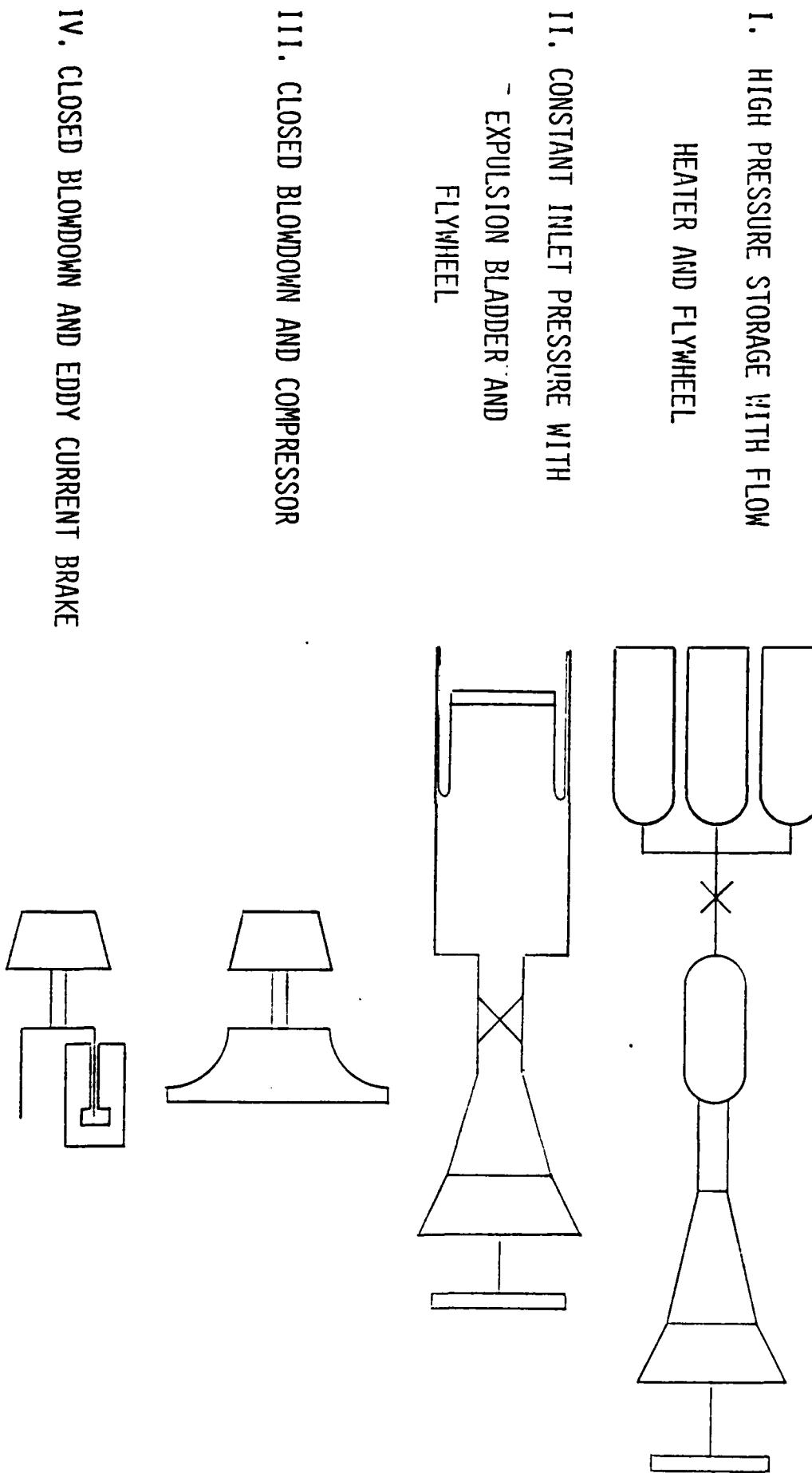


FIGURE 7

POWER BALANCETURBINE CONNECTED TO EDDY CURRENT BRAKETURBINE

$$\text{POWER PRODUCED} \sim M_{\text{TIP}}^2 \sim N^2$$

EDDY BRAKE

$$\text{POWER ABSORBED} \sim B_0^2 U^2 \sim N^2$$

CONCLUSIONS

- BRAKE & TURBINE CHARACTERISTICS MATCH
- NO CONTROL SYSTEM REQUIRED
- MAGNETIC FIELD STRENGTH FINE TUNES BRAKE

FIGURE 8

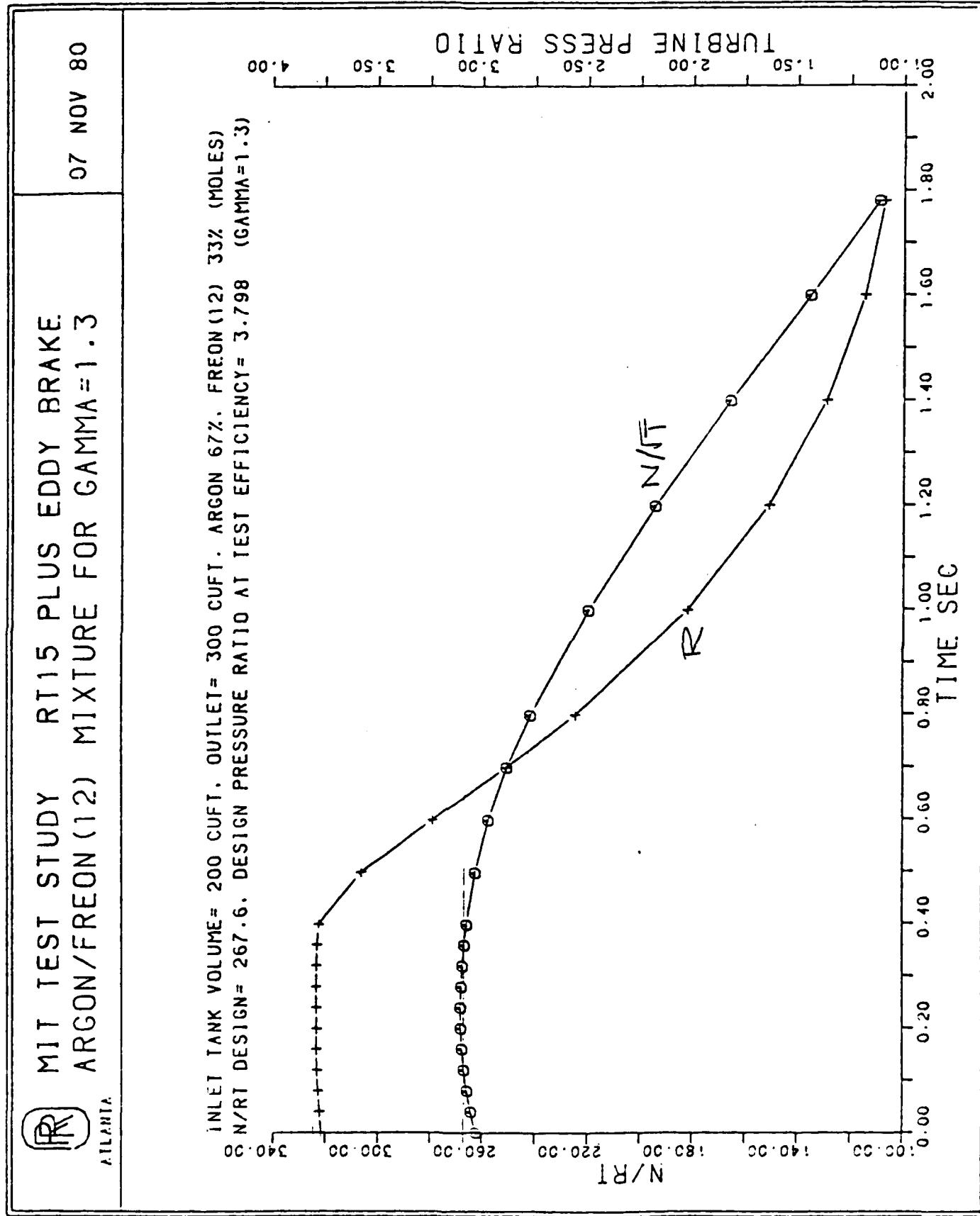


FIGURE 9

MAIN FLOW VALVE

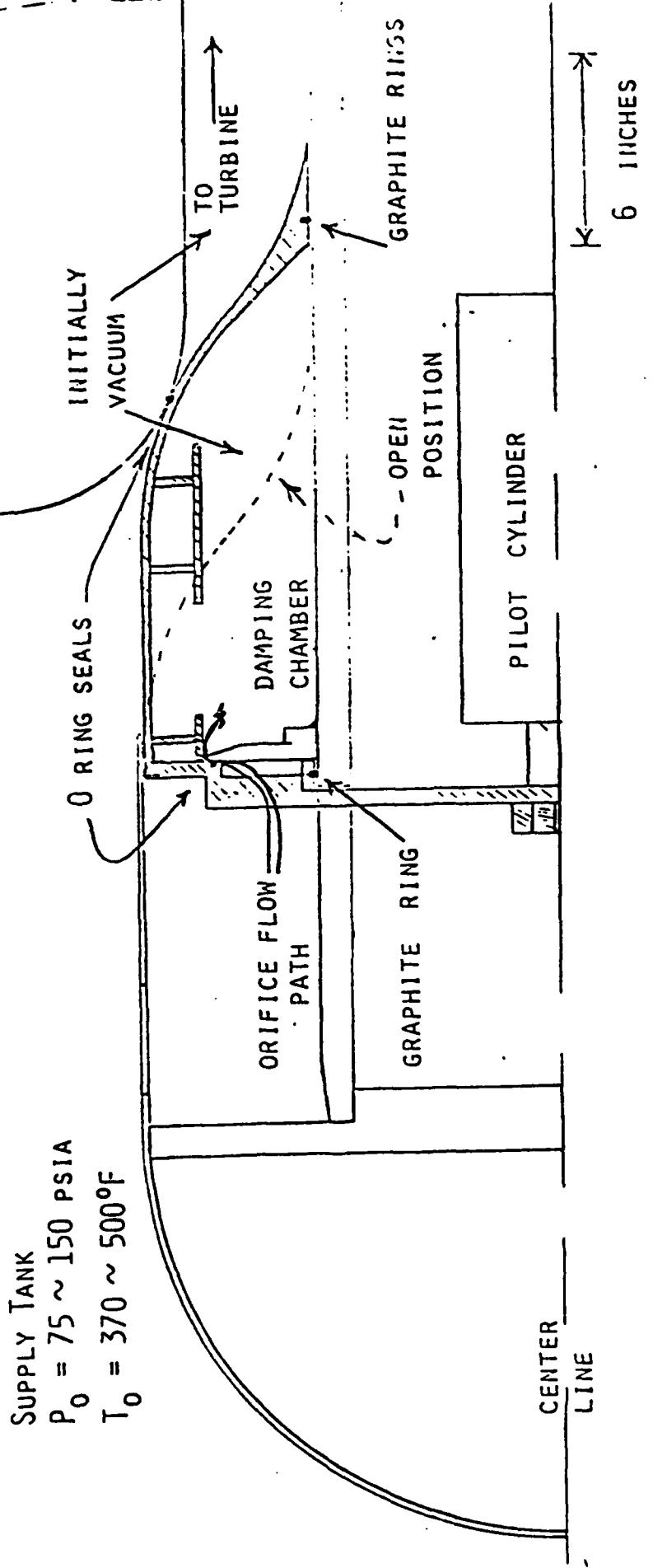
FACILITY REQUIREMENTS

- NONRESTRICTING - OPEN 2 FT. DIAMETER
- FAST ACTING - 50MS FULL OPEN
- HIGH TEMPERATURE OPERATION - UP TO 500°F
- EXCELLENT SEALING - VACUUM TO 150 PSIA

DESIGN IMPLEMENTATION

- ANNULAR PLUG DESIGN CHOSEN
- PILOT OPERATED - GAS FLOW PROVIDES OWN FORCE
- CARBON STEEL CONSTRUCTION - OIL HEATED
- ALL SEALS USE "O" RINGS

FIGURE 10



133

MAIN VALVE LAYOUT

FIGURE 11

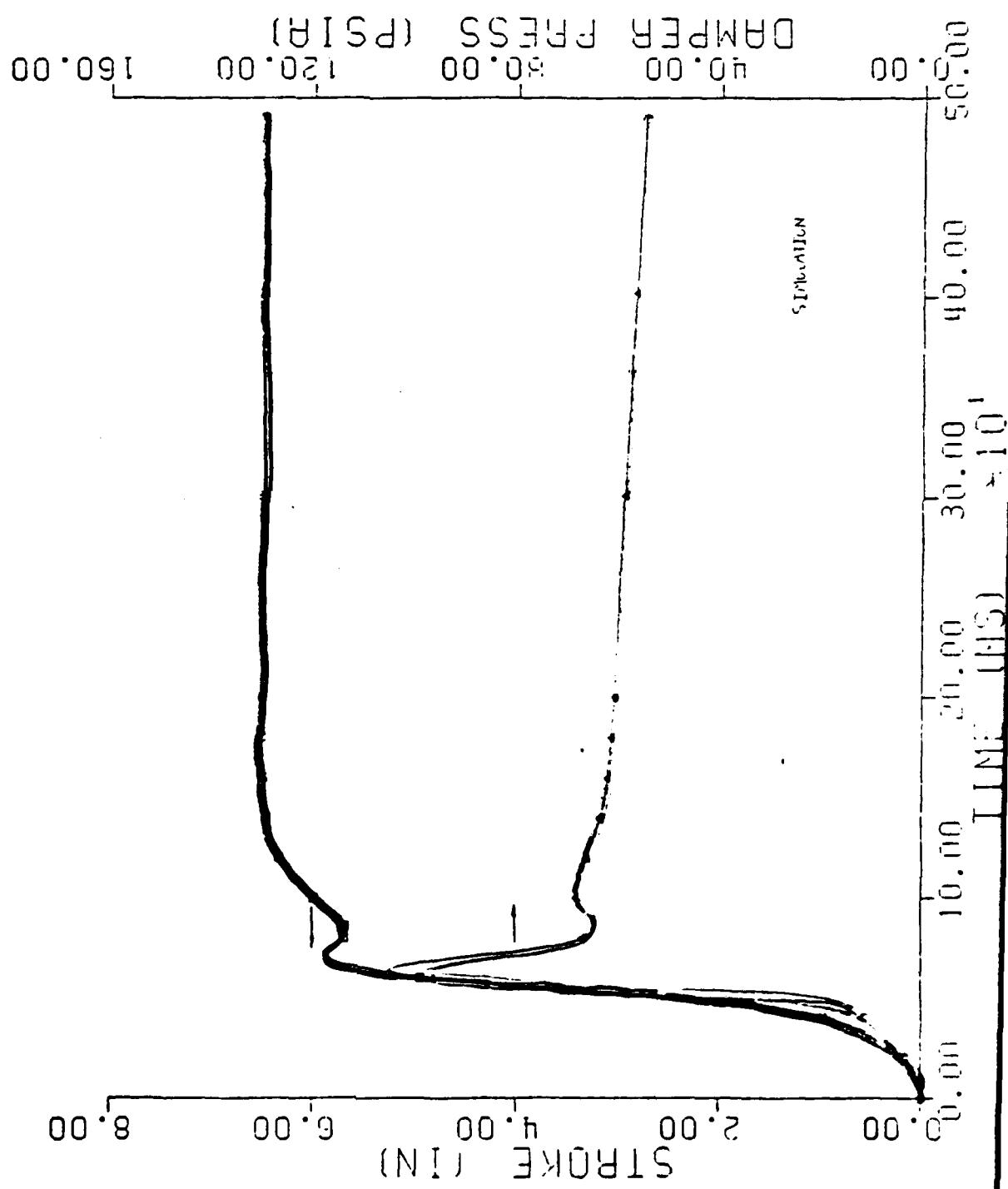


FIGURE 12
EDDY CURRENT BRAKE: TORQUE-SPEED CHARACTERISTIC

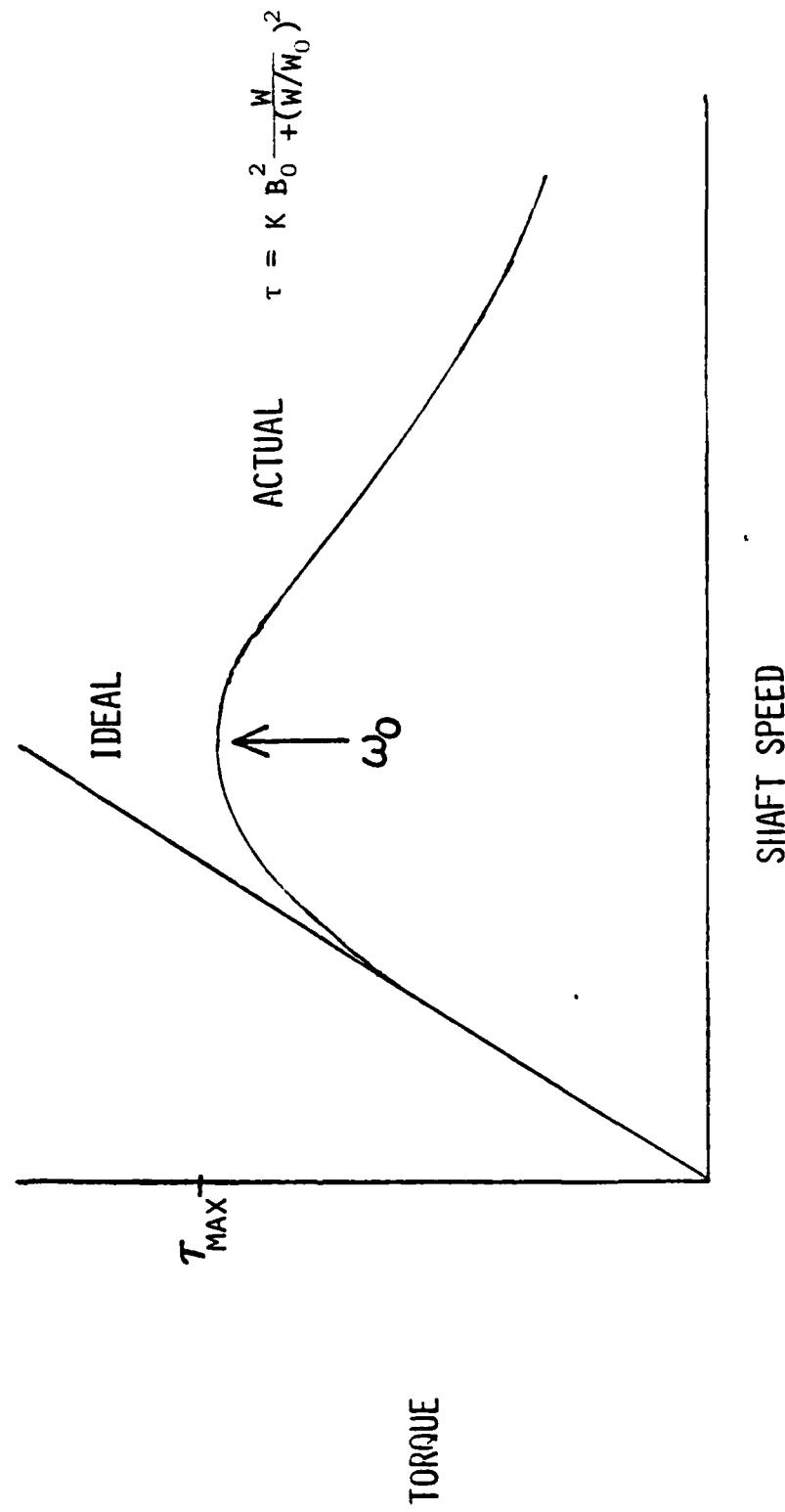


FIGURE 13

EDDY CURRENT BRAKE - DESIGN SUMMARY

Operating Point	$\omega = 662 \text{ rad/sec}$
	$P = 1.5 \times 10^6 \text{ watts}$
Drum Material	INCO 718, $\sigma = 0.801 \times 10^6 (\Omega\text{-m})^{-1}$
Geometry	Drum Radius, $r = .16 \text{ cm}$ Drum Thickness, $\Delta = .635 \text{ cm}$ Total Air Gap Height, $2g = 1.27 \text{ cm}$ Number of Poles, $N = 20$
Gap to Pitch Ratio	$\frac{2g}{\tau} = 0.25 \text{ (normal)}$
Resisting Factor	$(\frac{1}{k_r}) = 0.906$
Max. Torque	
Reynold's Number	$R_{M_0} = 0.7854 (\pi = \frac{2g}{\tau})$
Max. Torque Speed	$\omega_0 = 870 \text{ rad/sec (normal)}$
Power at Max. Torque	$P_0 = 2.045 \times 10^6 \text{ watts}$
Static Field	$B_0 = 0.781 \text{ T}$
Amp. Turns per Pole	$I = 7896 \text{ amps}$
Excitation Coils	
Time Constant	$\tau_c \text{ (sec)} \quad 0.010 \quad 0.020 \quad 0.025$
Total Power	$P_c \text{ (watts)} \quad 47,000 \quad 23,500 \quad 18,800$
"Copper" cross-section	$A_c \text{ (cm}^2\text{)} \quad 1.60 \quad 3.21 \quad 4.01$

FIGURE 14
BLOWDOWN TURBINE FACILITY

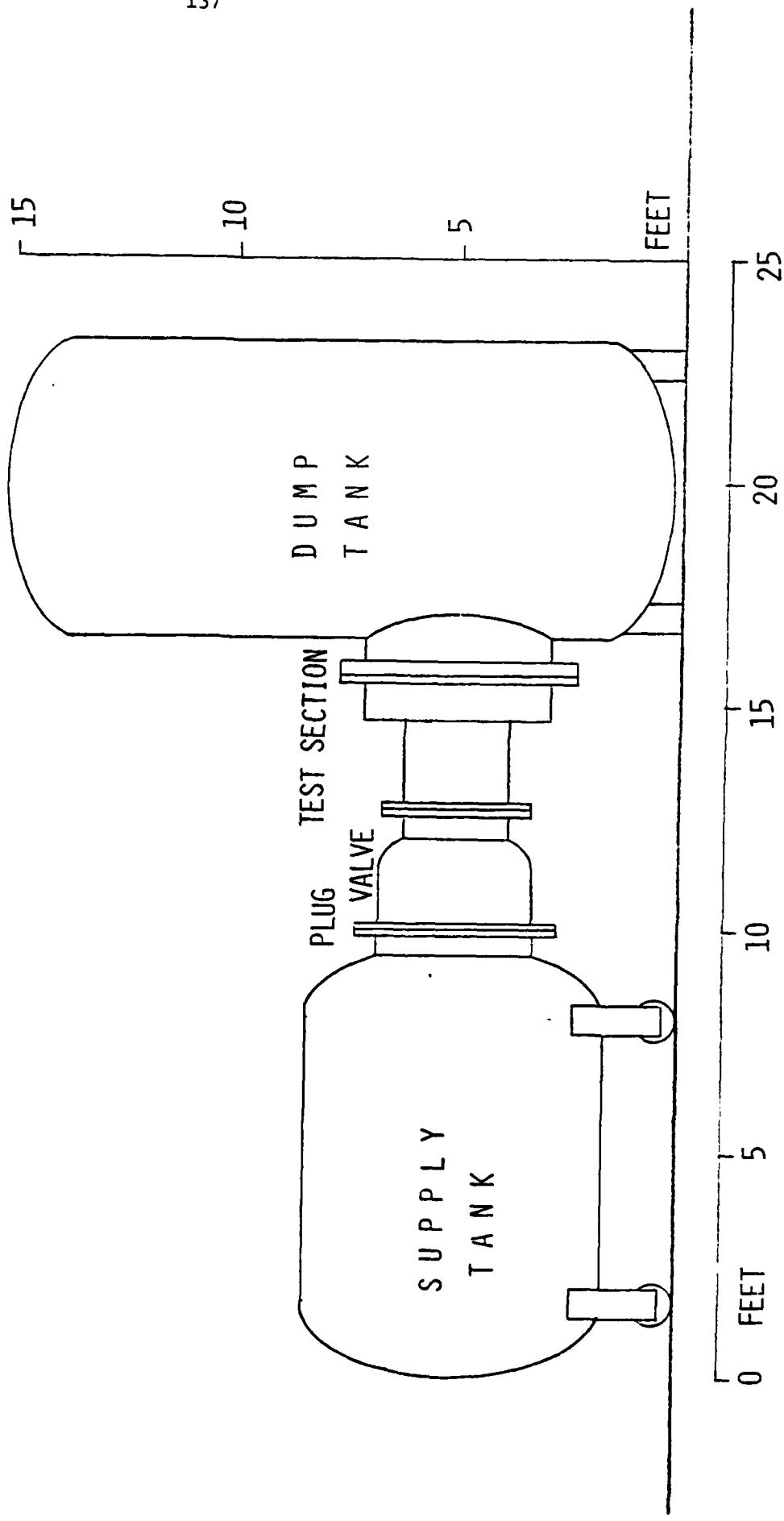


FIGURE 15

COMPUTER ROOM

BLOCK DIAGRAM

TRANSIENT ACQUISITION SYSTEM

TEST CELL

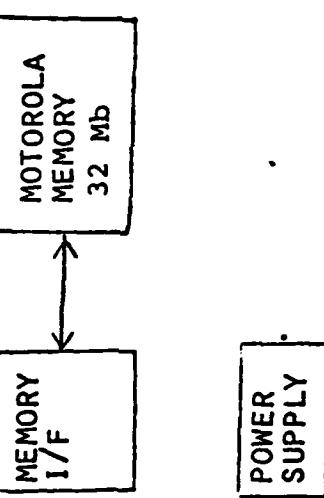
LINE
DRIVERS

200 ft

256 Mb/SEC

A/D
200KHZ/
CH

100 CHANNELS



139
FIGURE 16

DATA ACQUISITION SYSTEM COMPARISON

	<u>CUSTOM SYSTEM</u>	<u>BEST COMMERCIAL SYSTEM</u>
RESOLUTION/CH	12 BITS	10 BITS
MAX SAMPLING RATE	200 KHz	1000 KHz
APERTURE UNCERTAINTY	100 ps	100 ps
LINEARITY	0.024%	0.2%
ABSOLUTE ACCURACY	0.1%	1.0%
SAMPLING MODE	SIMULTANEOUS	SIMULTANEOUS
GAIN/OFFSET CONTROL	FRONT PANEL	PCB
CLOCK GENERATOR	4 SPEED; 20 Hz-200 KHz	3 SPEED; 20 Hz-20MHz
CLOCK RESOLUTION	INVERSE	GEOMETRIC
SAMPLE SETUP	COMPUTER	SWITCHES
MANUAL READOUT	COMPUTER DISPLAY	OSCILLOSCOPE
TOTAL SAMPLES/CH	400,000	1,000

FIGURE 17

PRESSURE-VELOCITY INSTRUMENTATION

- SENSOR TECHNOLOGY - SILICON DIAPHRAGM, STRAIN GAUGE
- WALL STATIC PRESSURE - CASING, NGV'S
- TRAVERSING PROBE - FLOW TOTAL & STATIC PRESSURE, 2 FLOW ANGLES
 - STATIONARY FRAME TRAVERSE
 - ROTATING FRAME TRAVERSE (BEHIND ROTOR)
- TECHNIQUES ADAPTABLE FROM COMPRESSOR RESEARCH (SIMILAR TEMPERATURES & PRESSURES)

¹⁴¹
FIGURE 18

HEAT TRANSFER INSTRUMENTATION (NGV'S AND ROTOR)

ALTERNATE TECHNIQUES AVAILABLE

A. BLADE CALORIMETER APPROACH

- USE BLADE AS CALORIMETER
- MEASURE TEMPERATURE WITH THIN FILM THERMOCOUPLES
- USE 3-D TRANSIENT HEAT TRANSFER CODE TO REDUCE DATA

ADVANTAGES

- MINIMUM FLOW DISTRUPTION
- INSTRUMENT INDUCED TEMPERATURE ERROR LOW
- SINGLE INSTRUMENTATION MEASURES HIGH & LOW FREQUENCIES

DISADVANTAGES

- REQUIRES 3-D TRANSIENT HEAT TRANSFER CODE
- LARGE CAPACITY DATA ACQUISITION SYSTEM NECESSARY
- CALIBRATION RELATIVELY DIFFICULT

B. DISCRETE CALORIMETER APPROACH

- DISCRETE CALORIMETERS INSERTED IN BLADES
- CALORIMETERS (1 MM DIAMETER) THERMALLY ISOLATED FROM BLADE

ADVANTAGES

- SIMPLIFIED DATA REDUCTION
- EASILY CALIBRATED

DISADVANTAGES

- DISTURBS BLADE BOUNDARY LAYERS
- INTRODUCES DOWNSTREAM ERRORS
- DIFFICULT TO MEASURE HIGH & LOW FREQUENCIES WITH SAME UNIT

FIGURE 18

HEAT TRANSFER INSTRUMENTATION (CONT'D)C. THIN FILM HEAT FLUX GAUGES APPROACH

- MULTI LAYER FLUX GAUGES FABRICATED ON BLADE SURFACE
- DIFFERENTIAL TEMP MEASUREMENT ACROSS INSULATOR

ADVANTAGES

- MINIMUM FLOW DISRUPTION
- INSTRUMENT INDUCED ERROR LOW
- DIRECT READOUT OF HEAT FLUX

DISADVANTAGES

- MUST BE VERY THIN FOR HIGH FREQUENCY RESPONSE (1 TO 5 μ M)
- DIFFICULT TO FABRICATE

FIGUPE 19

THIN FILM HEAT FLUX GAUGEDESIGN

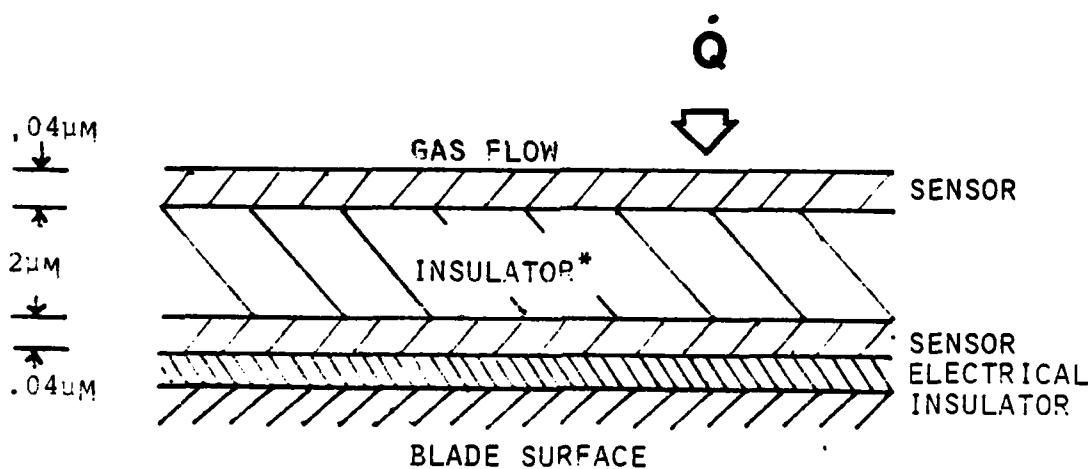
- MEASURE TEMPERATURE DIFFERENCE ACROSS INSULATOR
- SENSORS ARE TEMP. SENSITIVE RESISTORS

CONSTRAINTS

- FREQ. RESPONSE LIMITS INSULATOR THICKNESS
- SELF HEATING SETS SENSOR RESISTANCE

TRADEOFFS

- SENSITIVITY FOR FREQ. RESPONSE
- RESISTANCE FOR TEMP. COEF.



*NOT TO SCALE

END

FILMED

6-83

DTIC